Load
REVISTA MENSUAL PARA USUARIOS DE
MARZO 1986
Nº 1 #2

sinclair

Programas

INEDITOS PARA CZ 1000 1500 2000 TK 83 85 90X

APLICACIONES COMERCIALES Y EDUCATIVAS

Z-80: EL REY DE LOS PROCESADORES

GENESIS DE LA PROGRAMACION

Gráficos

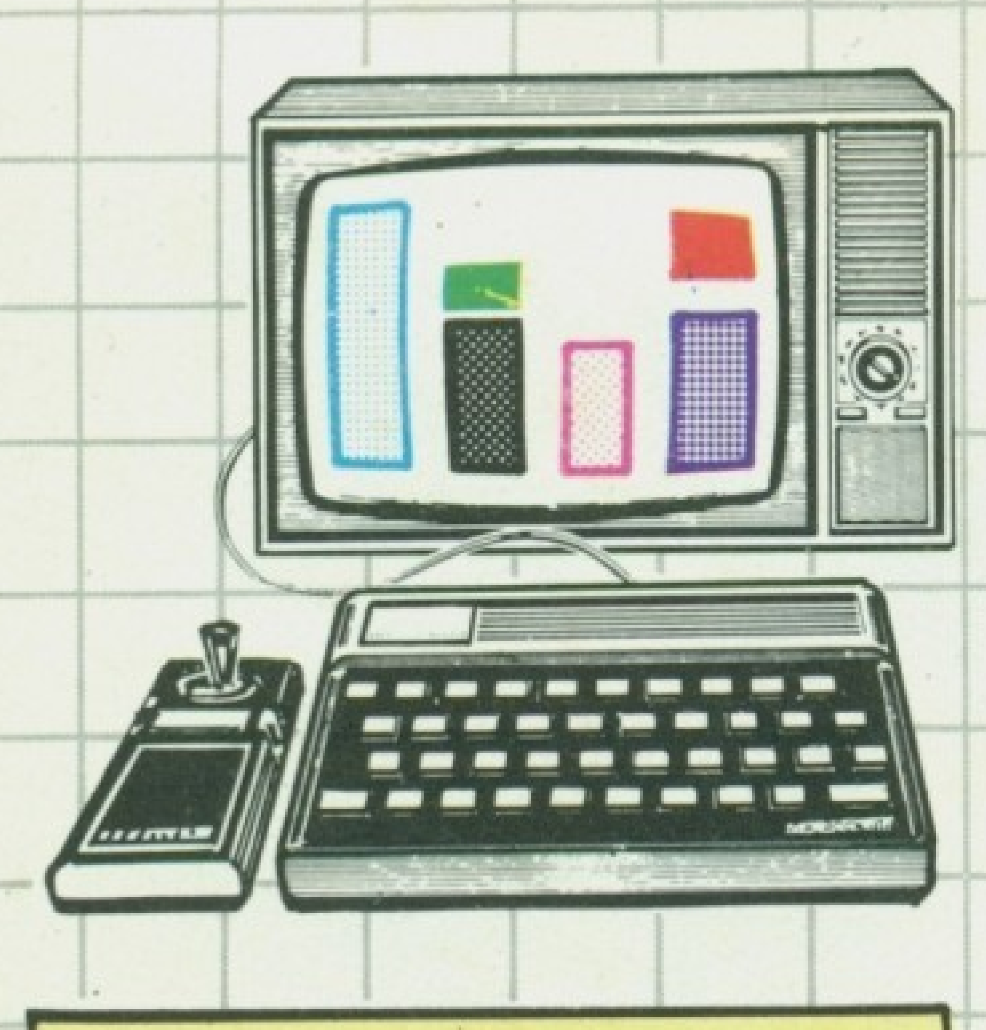
LOS MONSTRUOS MAS FAMOSOS

MICROCOMPUTADOR MICRODIGITAL TK-90X

Color y sonido a través del T.V. 16K y 48 K



EL MICROCOMPUTADOR CON MILES DE PROGRAMAS



GARANTIA 6 MESES

En venta en comercios de microcomputadores, artículos del hogar, electrónica, fotografía y librerías.

SOFTWARE Y PERIFERICOS TOTALMENTE COMPATIBLES CON ZX SPECTRUM +" ®

- Control del volumen del sonido a través del TV (sintetizador operado por BASIC)
- Interface incorporado para joystick
- Mensajes de ejecución y código de reportes de errores en castellano.
- TRACE: Comando de seguimiento de programas, permitiendo la rápida corrección de errores de lenguaje.
- UDG: Comando de editor de caracteres especiales definidos por el usuario (acentos, Ñ, etc.).
- Feedback sonoro del teclado
- Fuente de alimentación con interruptor.
- Ameno, fácil y completo manual de instrucciones en castellano.

MICRODIGITAL

Importa y distribuye: ARVOC s.a.i.c.f.i.
Tte. Gral. J. D. Perón 1563 (Ex Cangallo) (1037)
Capital Federal – Tel.: 35 - 2400/2511/8241.

STAFF

AÑO 1 - Nº 1 MARZO 1986

Director General
Ernesto del Castillo

Director Editorial
Cristian Pusso

Director Periodístico
Fernando Flores

Director Financiero
Javier Campos Malbrán

Arte y Diagramación
Fernando Amengual

Coordinador
Ariel Testori

Redacción
Eduardo Mombello

Departamento de Avisos
Oscar Devoto

Departamento de Publicidad
Guillermo González Aldanur

LOAD Revista para usuarios de SINCLAIR es una Revista mensual editada por Editorial PROEDI S.A., Cerrito 1320, 1° Piso, Buenos Aires, Tel.: 42-9681/9. Registro Nacional de la Propiedad Intelectual: E.T. M. Registrada.

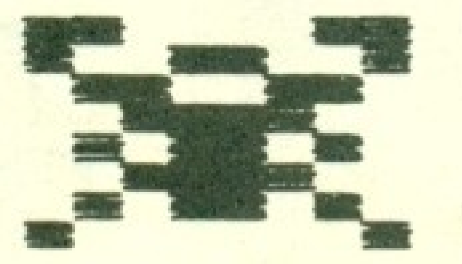
Precio de este ejemplar: A 2.
Impresión: Calcotam. Fotocromo tapa: Columbia. Fotocomposición: Van Waveren.
Los ejemplares atrasados se venderán al precio del último número en circulación.

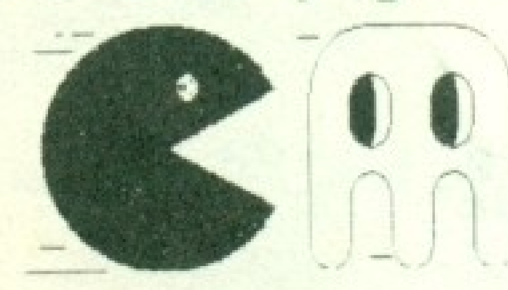
Distribuidor en Capital: Martino, Juan de Garay 358, P.B. Capital. Distribuidor interior: DGP: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800

MENU

New *pág. 4*
Las novedades del mercado informático internacional.

Los monstruos más famosos *pág. 6*



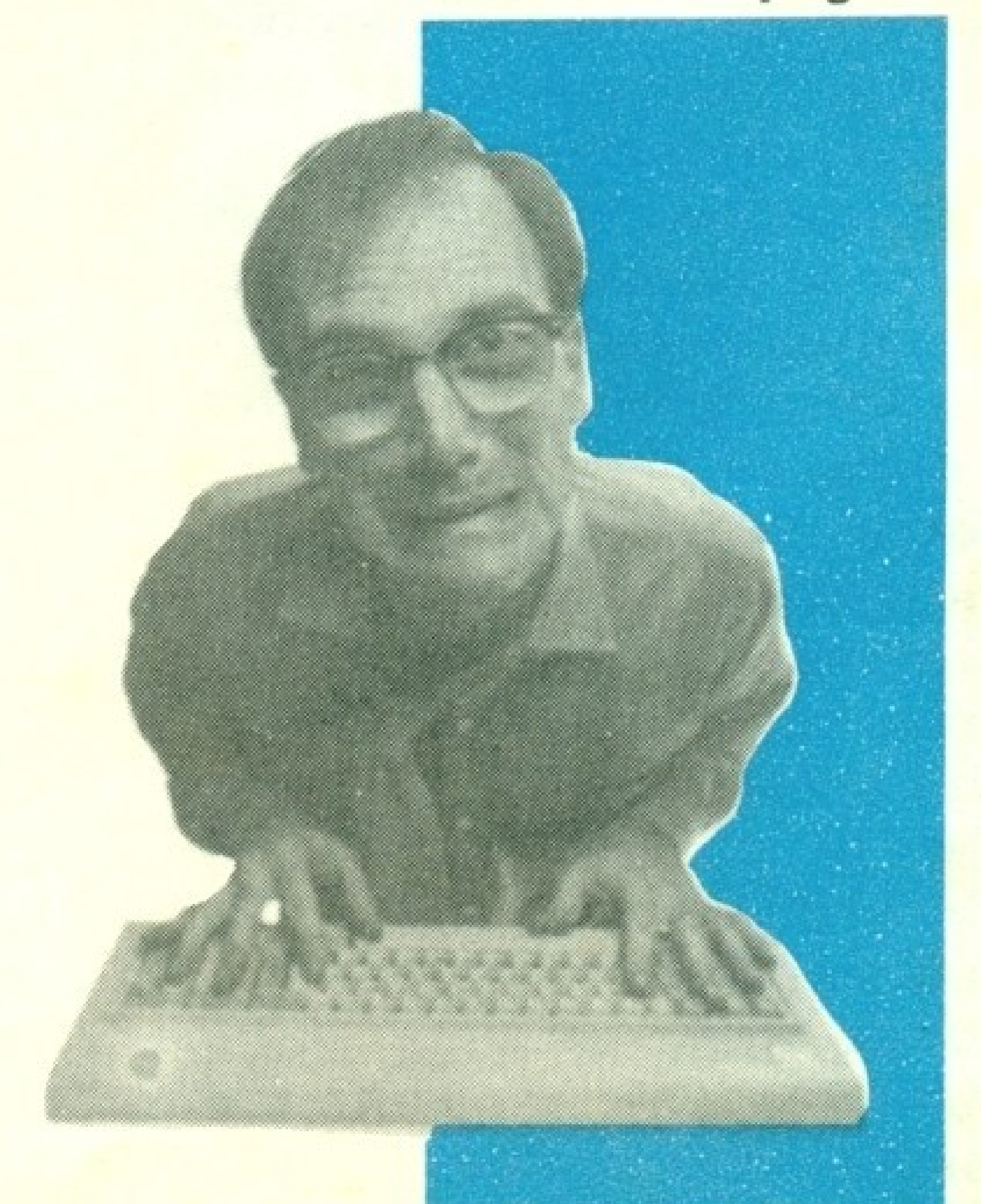


Les decimos cómo crear los gráficos de los personajes más famosos de los juegos, para que los puedan incluír en sus propios programas.

Z80: El rey de los procesadores *pág. 12*
Dentro del mundillo de las unidades centrales de proceso, con ocho bits de datos y dieciséis bits de direcciones, presentamos al Z-80 en una ficha técnica.

Génesis de la programación *pág. 16*
Explicamos las reglas para lograr un buen software, y que además sea comprensible para todos los que están en este apasionante mundo.

Matrices *pág. 20*



Este artículo va dirigido a estudiantes de los últimos años del ciclo secundario, o primer año de la universidad, y a todos aquellos que necesiten programar la carga y operación de distintos tipos de matrices.



Programas inéditos para
TK 83/85, CZ 1000/1500
- Music Machine (pág. 10)
- Destructor de estrellas (pág. 10)
- Interés compuesto (pág. 11)
- Factores (pág. 11)
- Archivo telefónico (pág. 14)
- Reducción de matrices (pág. 22)
- Fórmula 1 (pág. 32)

TK90X, TS 2068, CZ 2000
- Caja de ahorros (pág. 9)
- Control de tránsito (pág. 18)
- Dictanúmeros (pág. 25)

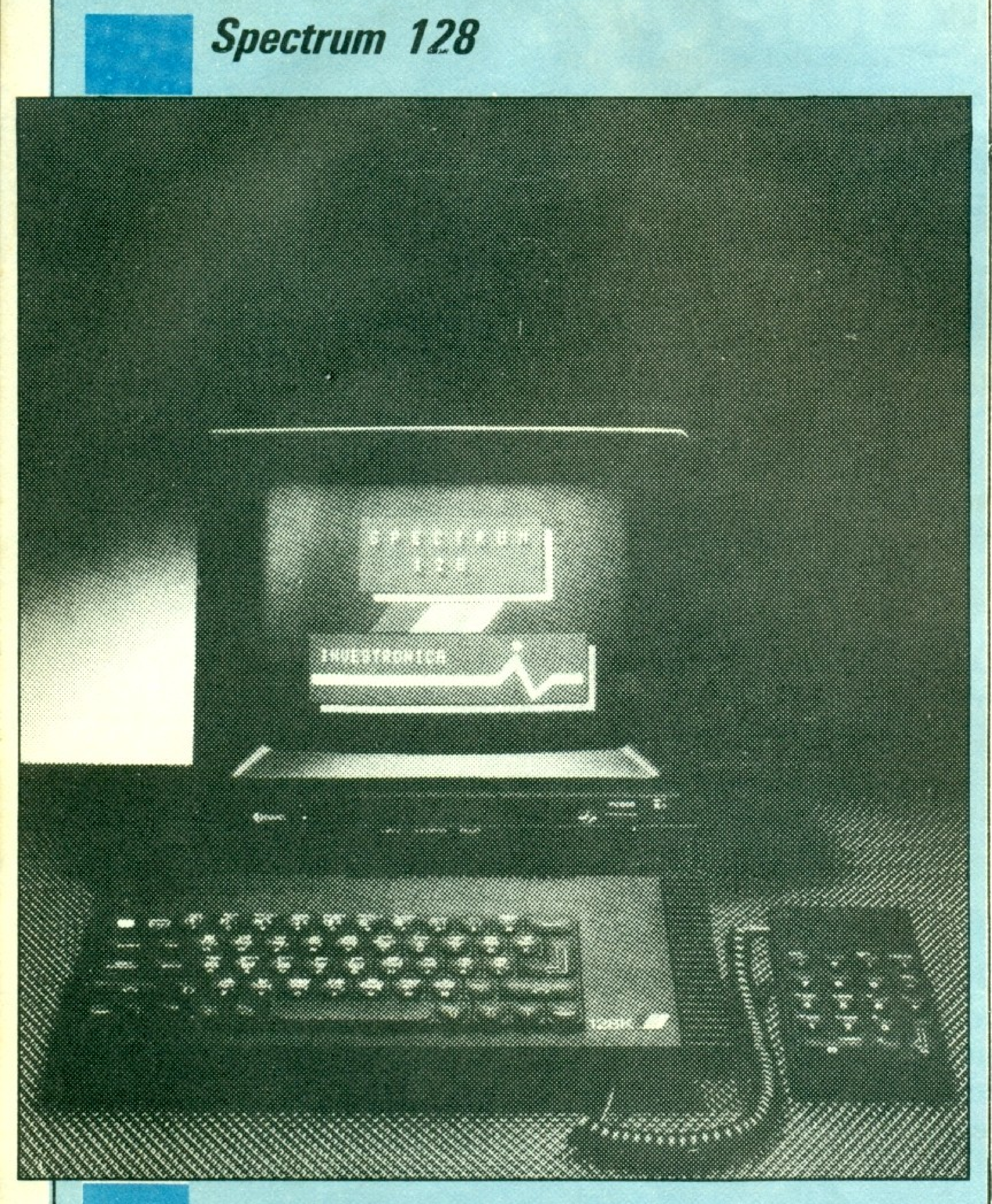
Software al día *pág. 5*
Comentamos Match Point y Psst.

Magic *pág. 15*
Engaños, hechizos y descubrimientos.

Un universo paralelo *pág. 28*
Explicamos el manejo de las interrupciones en el microprocesador Z-80.

Buzón *pág. 34*
Recibimos consultas y sugerencias, y las contestamos.

Spectrum 128



Sir Clive Sinclair junto con la compañía española Investrónica ha lanzado en ese país un nuevo primo para el ya famoso Spectrum de 48 K.
El nuevo Spectrum con 128 K y apariencia de Plus, que se fabricará en Madrid antes que en Inglaterra, trae algunas bellezas que por ahora disfrutarán sólo algunos papis españoles.

Entre ellas se destaca su nuevo editor de pantalla incorporado en ROM, que permite la edición de programas en cualquier lugar de ésta, así como introducir comandos directamente.

Trae un teclado numérico independiente con teclas de cursor que permiten su utilización como calculadora. Es también posible, por medio de éste, editar programas.
Su conexión con el mundo exterior se amplía por medio de tres interfaces: RS 232, MIDI (para instrumentos musicales) y salida RGB de monitor.
Por medio de un comando permite que sea utilizada como su antecesora de 48 K, haciéndolas compatibles en este modo a lo que software se refiere.
Y de yapa 128 K de memoria.
Se calcula que en el país europeo su precio será de alrededor de 60000 pesetas.
Vaya uno a saber cuánto costaría aquí, si algún arriesgado importador se dignara a traerla...

Microordenador en forma de manzana

Se trata de un proyecto de la Universidad de Michigan (EE. UU), quienes han desarrollado un micro que tiene la forma y el tamaño de una manzana.
El ordenador se cuelga de un árbol y sigue el mismo proceso que cualquier otra manzana:
se recoge, empaqueta y se envía al detallista. De esta forma se trata de seguir el proceso hasta el final, a fin de analizar las variaciones de humedad y temperatura, y averiguar de esta forma las causas que dañan la fruta.
Los ingenieros e investigadores de la Universidad de Michigan afirman que en base a simulaciones hechas por computadora, todas las funciones que ha de cumplir este frutal ordenador, han sido probadas con éxito, salvo la última... la extracción del juguito.

Interfase

La empresa inglesa ITL Kathmill, anuncia la aparición de un interface que permitirá conectar a una serie de ordenadores hasta cuatro unidades de disco de cualquier tipo, ya sea de 3, 3-1/2 ó 5-1/4 pulgadas a su ordenador.
Esta interface está lista en sus versiones para el Commodore 64, Oric y se está preparando la versión para el Spectrum.
Una de las mayores ventajas es que permitirá el uso de programas bajo el sistema operativo CP/M, en cualquiera de estos ordenadores. Como ventaja adicional para los usuarios del Spectrum, el interface incorpora 16 K de RAM para que el ordenador tenga los 64 K requeridos para el CP/M, además de un interface RS 232 y otro del tipo Centronics.
La misma empresa fabrica su propia unidad de discos de 3 pulgadas compatibles con el sistema. Los discos son de doble cara doble densidad permitiendo una capacidad de almacenamiento de 220 K por cara una vez formateado el disco.
En fin, alguna vez será...

PSSST

ORDENADOR: CZ2000 Spectrum/TK 90
SOPORTE: Cassette
CLASE: Juego
CONFIGURACION: 16 K

"No amarás a los insectos que se comen las plantitas", es el lema de este juego.

En efecto, nos transformaremos en jardineros y nuestro trabajo será cuidar una planta un tanto grande que crece (si no se le prende ningún bicho), en el centro de la pantalla, y parece ser un girasol.

Para lograr esto contamos con tres aerosoles, colocados en distintas partes de las paredes que lindan con el jardín; uno para cada clase de insecto. Si usamos un aerosol equivocado para matar, por ejemplo, a los gusanos, no le haremos ni cosquillas. Estos abominables bichos, no se contentan con chuparse nuestra planta, sino que si además nos tocan, nos quitarán vidas.

Pero no todas son pálidas; algunos simpáticos elementos aparecerán en las paredes, como regaderas, mata moscas, etc., que de recogerlos aumentará nuestro puntaje.

El cambio de nivel se producirá cada vez que nuestra plantita florezca, presentando a nuevos personajes malignos.

Los movimientos del jardinero están controlados por las siguientes teclas:

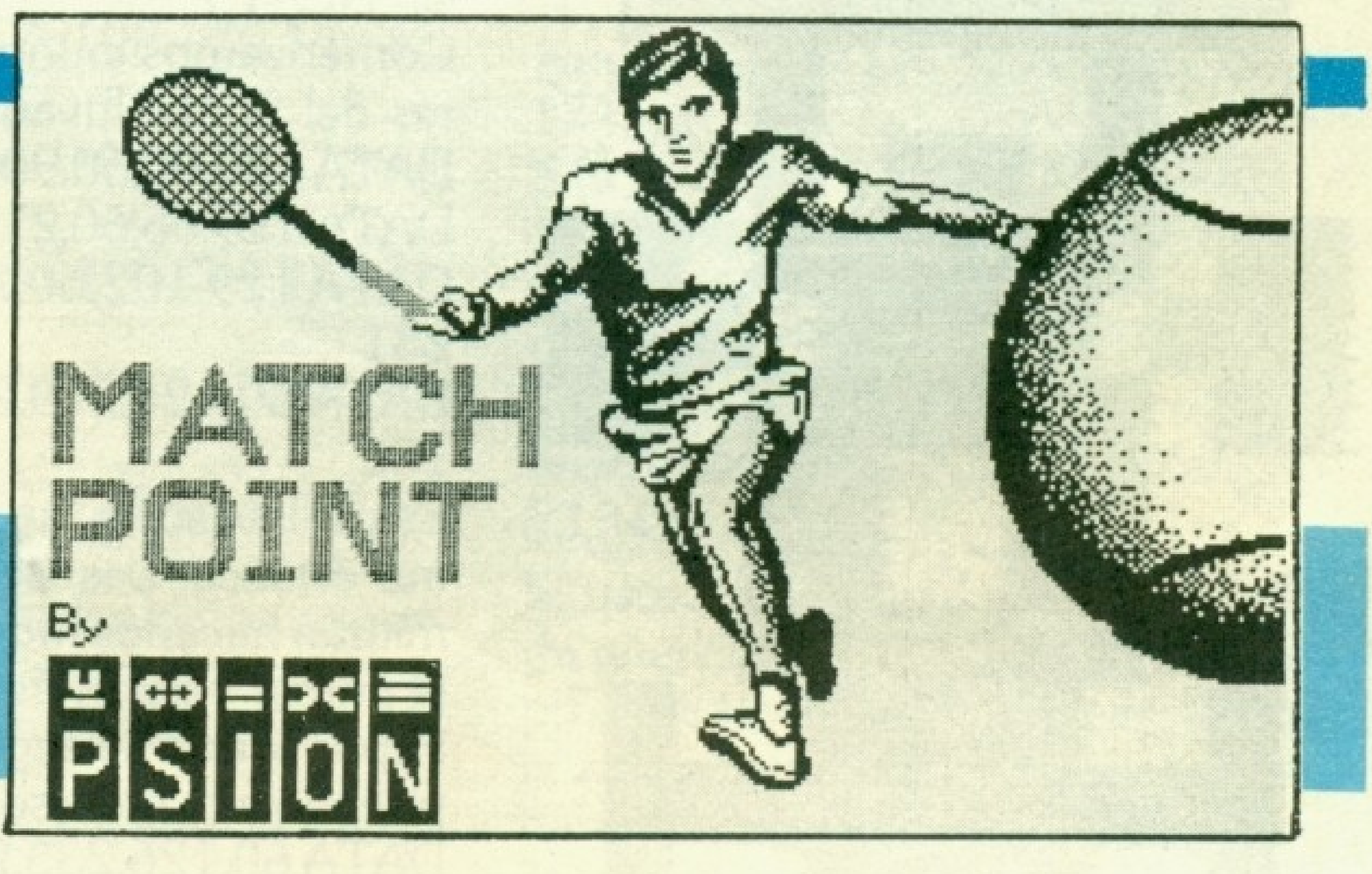
Q izquierda
W derecha
E abajo
R arriba
T rocío

El menú de comienzo presenta las siguientes opciones:

1 un jugador
2 dos jugadores
3 teclado
4 Kempston joystick
5 comienzo

Es un juego en el que están muy bien aprovechados los 16 K disponibles. Tiene un desarrollo gráfico y de movimientos muy bueno, sonidos aceptables, y se ha logrado un buen nivel de adicción.

MATCH POINT



ORDENADOR: TK 90/CZ2000 Spectrum
SOPORTE: Cassette
CLASE: Juego
CONFIGURACION: 48 K

Si no nos gustan los grandes torneos de tenis, demos vuelta la página, pues éste es uno de los mejores programas de simulación de juegos tradicionales que existe en el mercado argentino (por lo menos).

Con la opción de jugar contra la máquina, o versus un amigo, este soft muestra todas las posibles combinaciones de opciones que se pueden efectuar, en el menú de comienzo.

Las opciones separadas por grupos son:

Cuartos de final
Semi final
Final
Exhibición
Un jugador
Dos jugadores
Un set
Tres sets
Cinco sets
Sinclair joystick
Kempston joystick
Teclado

Por medio de la tecla BREAK elige uno de los grupos anteriores y con el 0 una de las tres opciones de ese submenú, para luego comenzar el juego pulsando ENTER.

La pantalla nos da una visión de fondo y un poco elevada de la cancha. Nuestro nombre y el del adversario aparecen junto a los contadores de Game, Set y Puntos, frente a nuestra ubicación.

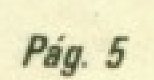
Este juego cumple con todas las reglas del tenis actual, y también nos las hace cumplir. No hay que preocuparse por no tener un buen revés o saque potente, pues aquí seremos expertos en estas cosas; la dificultad radica en correr detrás de la pelotita y por supuesto pegarle.

Tampoco debe desvelarnos la idea de que la pelota se cuelgue en la red, pues un enanito muy simpático de piernas rápidas, la recogerá muy amablemente.

Un público enardecido seguirá muy atentamente el juego; no nos dejemos distraer...

Solamente tres cosas no fueron previstas por sus programadores; no podemos usar vincha, ni arrojar la raqueta y no nos harán parte del Jet Set.

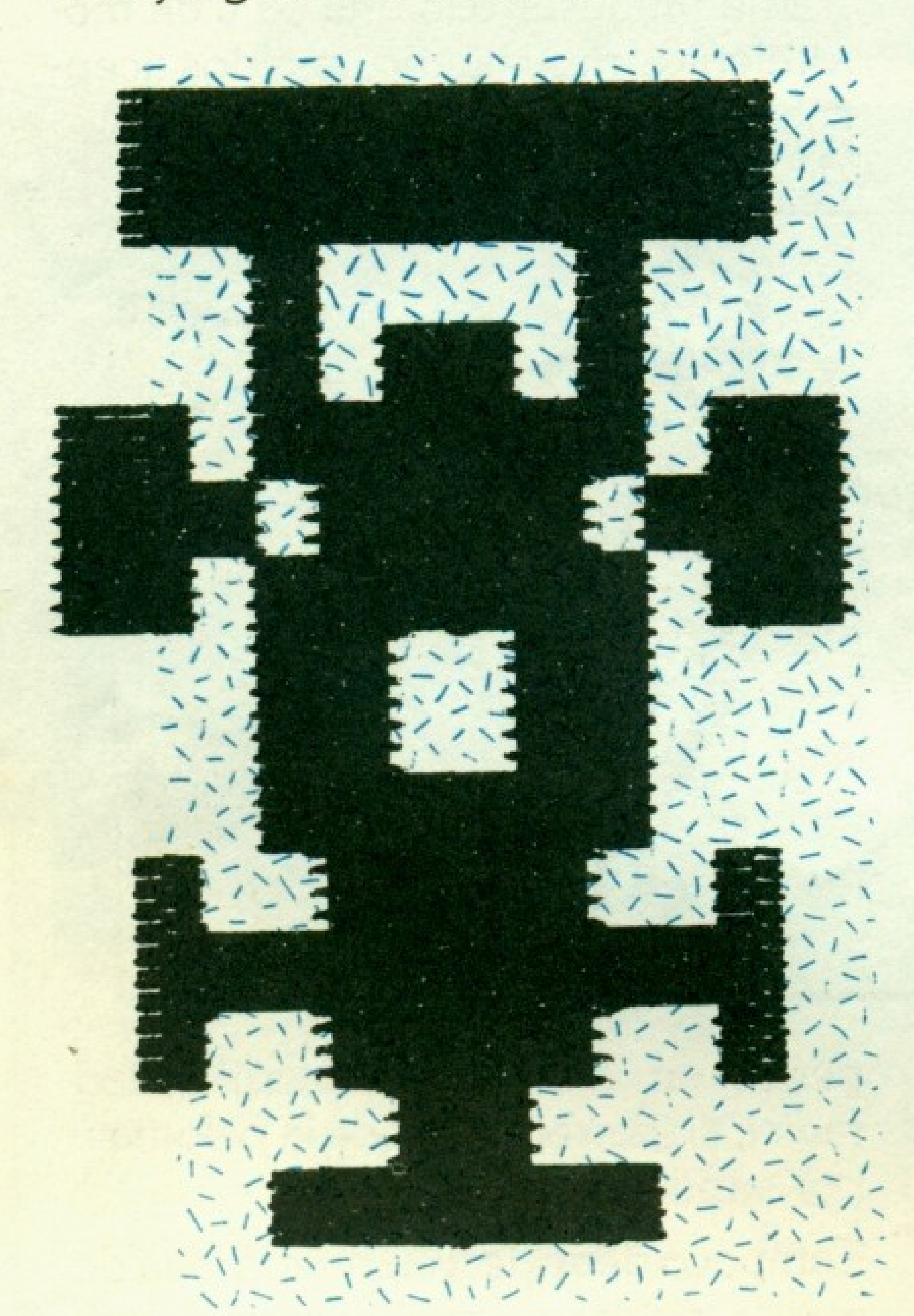
Por lo demás, podrá tener un lúcido juego de fondo, hacer hermosos globitos y todos los chiches clásicos de este juego (si puede...).





Les presentamos los datos necesarios para crear los personajes más conocidos de los juegos de manera que los puedan incorporar a sus programas. Los gráficos están incluidos en instrucciones DATAs de 8 bytes de longitud. Así aparecerán desde los invasores espaciales hasta el famoso "Pac-Man"

Presentamos aquí una serie de DATAs cuyos números han de representar (en la zona de gráficos definibles por el usuario) a algunos de los personajes y protagonistas, de los video-juegos más famosos.



Les acercamos también algunos gráficos, que sin ser tan monstruosos, pueden despertar su imaginación. Cualquiera de estos DATAs revelará su identidad, si le asociamos un programa simple:

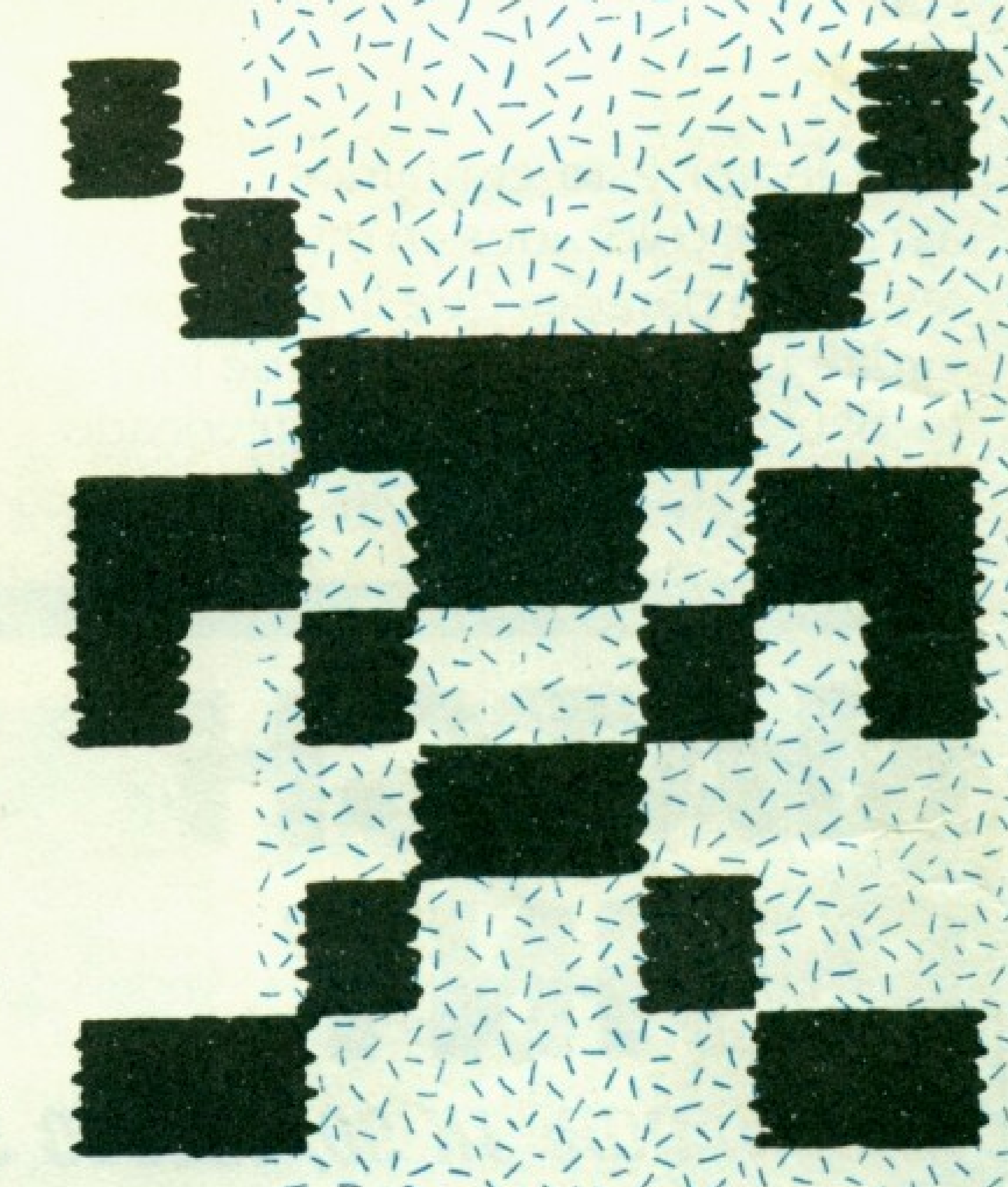
```
10 FOR F =0 TO 7; READ A
20 POKE USR "A" +F,A
30 PRINT AT 10,10;"A"
40 NEXT F
```

Comenzamos entonces con los actores del Space Invaders:

```
DATA 129,90,60,90,60,102,153,195
DATA 129,66,60,219,165,24,36,195
DATA 129,102,60,255,90,255,129,231
DATA 60,126,219,219,255,126,66,102
```

Estos "invasores" se presentan en forma estática. Los siguientes nos permitirán hacerlos mover, imprimiendo la primer versión del mismo, y sobreimprimiendo su segunda versión, repitiendo este proceso alternativamente.

```
DATA 60,126,235,235,255,126,36,231
DATA 60,126,215,215,255,126,66,102
DATA 66,36,189.173.255.60.36.231
```



```
DATA 36,36,60,247,189,189,36,36
DATA 129,189,189,173,239,60,36,231
DATA 231,165,189,181,52,255,129,231
```

DATA 0,24,60,90,126,24,36,90
DATA 60,90,126,60,24,36,90,0
DATA 198,34,20,42,28,34,17,230
DATA 99,68,40,84,56,68,136,103

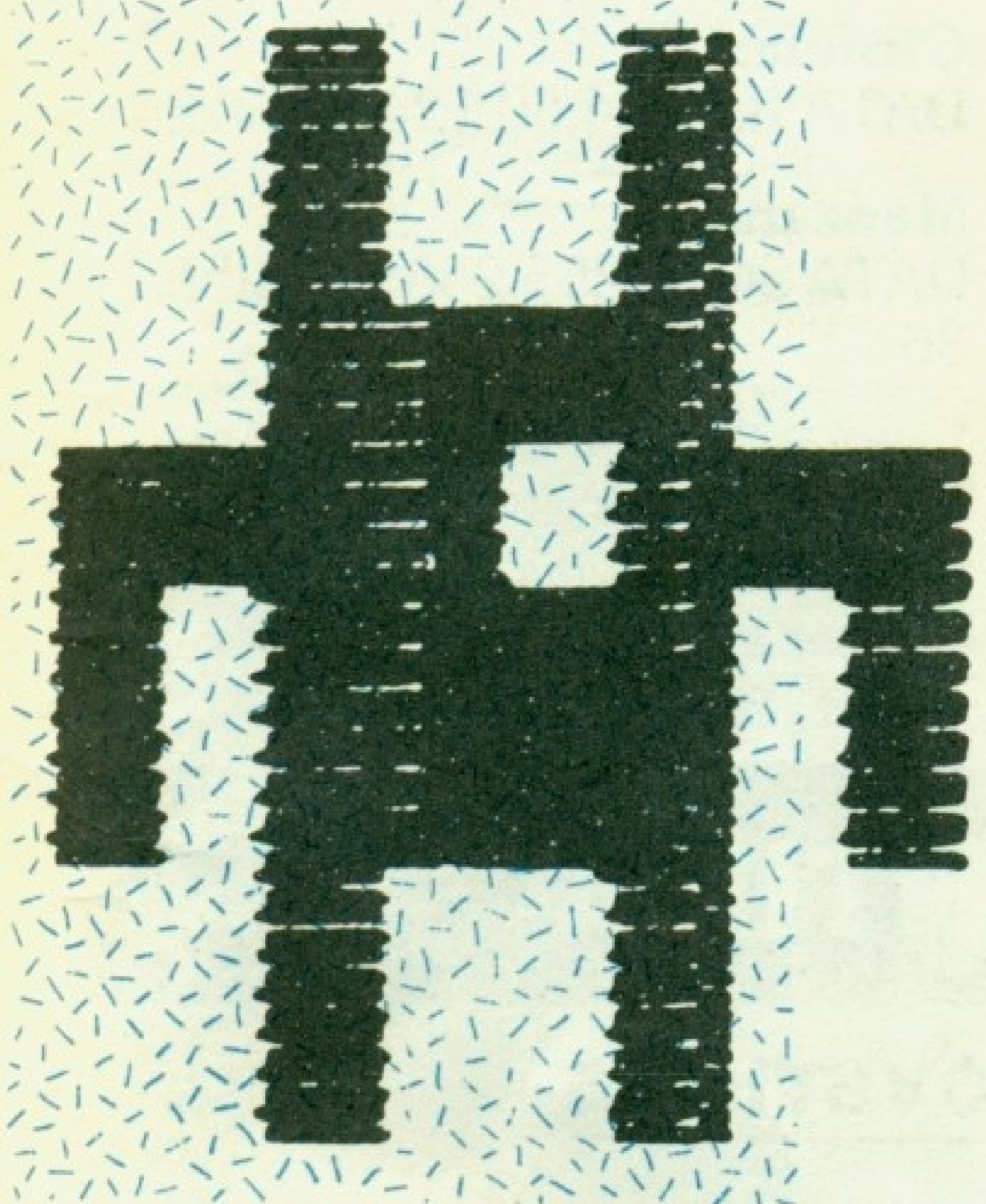
Lo que sigue es la representación estática y en movimiento del "Spacey", platillo volador bien conocido por los fans del Space Invaders.

DATA 60,126,255,170,170,255,126,60

Movimiento hacia la izquierda:
DATA 3,7,15,10,10,15,7,3

Movimiento hacia la derecha:
DATA 192,224,240,160,160,240,224,192

Ahora veremos tres "Spaceys" (de dos caracteres de largo cada una). Un buen efecto de movimiento se logra-



rá imprimiendo cada nave en el orden del listado, especialmente si se hace con velocidad.

Nave 1:
Izquierda: DATA 63,127,255,73,73,255,127,63
Derecha: DATA 252,254,255,36,36,255,254,252

Nave 2:
Izquierda: DATA 63,127,255,36,36,255,127,63
Derecha: DATA 252,254,255,146,146,255,254,252

Nave 3:
Izquierda: DATA 63,127,255,146,146,255,127,63
Derecha: DATA 252,254,255,73,73,255,254,252

Las que siguen son algunas bases de misiles:

DATA 24,24,24,60,126,255,219,219
DATA 0,24,24,24,255,255,255,255
DATA 24,24,24,60,255,255,255,0
Izquierda: DATA 1,1,7,9,31,127,255,255
Derecha: DATA 0,0,192,32,240,252,254,254

Aquí van algunas naves espaciales que nos ayudarán en la hora de elegirlas, para nuestros programas.

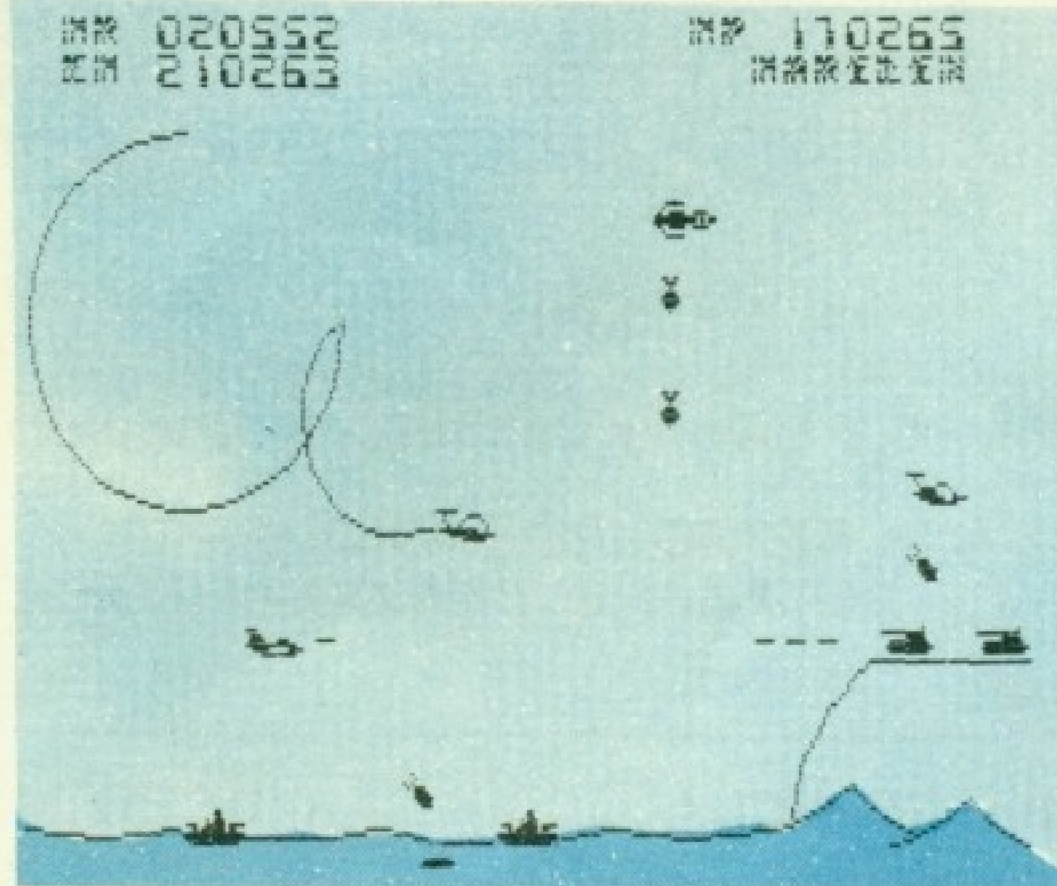
Izquierda: DATA 0,252,32,33,18,127,15,1
Derecha: DATA 0,0,240,8,4,228,255,248
Izquierda: DATA 31,32,79,255,255,79,32,31
Derecha: DATA 0,0,60,235,235,60,0,0
Izquierda: DATA 0,248,32,16,15,63,15,3
Derecha: DATA 0,0,0,224,16,8,207,248

Naves espaciales del estilo "Asteroid", son las que siguen. Estas están basadas en dos diseños, uno horizontal/vertical, y el otro diagonal.

Vuelo noroeste: DATA 192,176,76,67,44,40,16,16
Vuelo oeste: DATA 0,7,25,98,132,98,25,7
Vuelo sudoeste: DATA 16,16,40,44,67,76,176,192
Vuelo sur: DATA 198,170,146,68,68,40,40,16
Vuelo sudeste: DATA 8,8,12,44,194,50,13,3
Vuelo este: DATA 0,224,152,70,33,70,152,224
Vuelo noreste: DATA 3,13,50,194,44,12,8,8
Vuelo norte: DATA 16,40,40,68,68,146,170,198

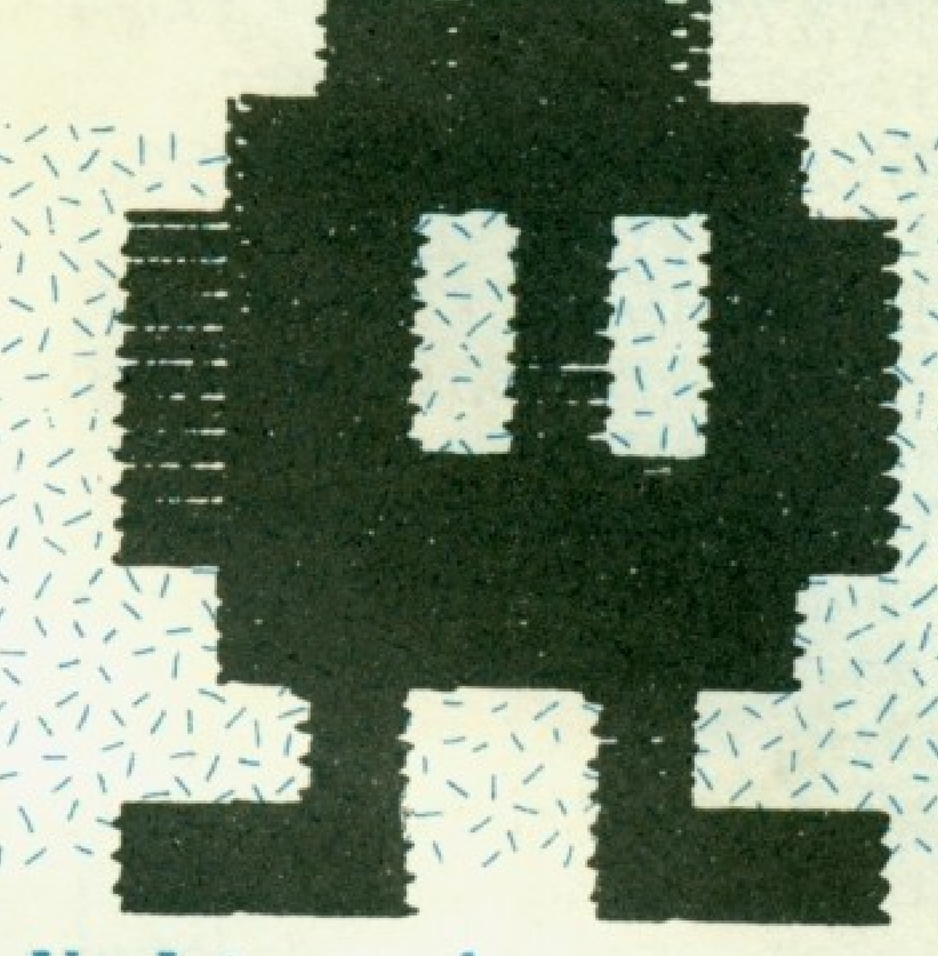
Los siguientes son los datos correspondientes a tres aviones de dos caracteres de largo.

Un Jet:
Izquierda: DATA 0,0,224,112,120,255,60,15



Derecha: DATA 0,0,0,0,224,248,30,248

Uno a hélice:
Izquierda: DATA 192,192,227,158,127,0,0,0
Derecha: DATA 0,0,96,248,252,224,240,112



Un Jet sesgado:
Izquierda: DATA 14,199,224,255,127,1,0,0
Derecha: DATA 0,0,96,248,252,224,240,112

Las siguientes ocho listas de datos, corresponden al gráfico de un avión, en cada una de las posiciones de la brújula o compás.

Norte: DATA 8,8,28,62,127,8,8,28
Noreste: DATA 1,2,124,60,28,172,68,32
Este: DATA 16,24,156,255,156,24,16,0
Sudeste: DATA 32,68,172,28,60,124,2,1
Sur: DATA 28,8,8,127,62,28,8,8
Sudoeste: DATA 4,34,53,56,60,62,64,128
Oeste: DATA 8,24,57,255,57,24,8,0
Noroeste: DATA 128,64,62,60,56,53,34,4

Si usted es un amante de los juegos de guerra y tiene la idea de hacer su propia batalla por tierra y agua, aquí le ayudamos con algunos tanques y barcos de combate.

Izquierda: DATA 0,16,31,31,63,127,63,0
Derecha: DATA 0,0,252,0,240,248,240,0
Izquierda: DATA 0,0,63,0,15,31,15,0
Derecha: DATA 0,8,248,248,252,254,252,0
Izquierda: DATA 0,0,2,115,23,255,126,63
Derecha: DATA 192,192,224,239,236,255,255,254
Izquierda: DATA 3,3,7,247,55,255,255,127
Derecha: DATA 0,0,64,204,232,255,254,252

Y si piensa en tener una vista superior del campo de batalla, nada mejor que tener los ocho movimientos de su tanque, como lo vimos previamente con los aviones.

DATA 8,73,73,93,127,93,65,65
DATA 33,66,188,60,60,29,2,4
DATA 254,16,56,63,56,16,254,0
DATA 4,2,29,60,60,188,66,33
DATA 65,65,93,127,93,73,73,8
DATA 32,64,184,60,60,61,66,132
DATA 127,8,28,252,28,8,127,0
DATA 132,66,61,60,60,184,64,32

Como un jueguito de estos sin bombas no es un juego, aquí van un par de ellas:

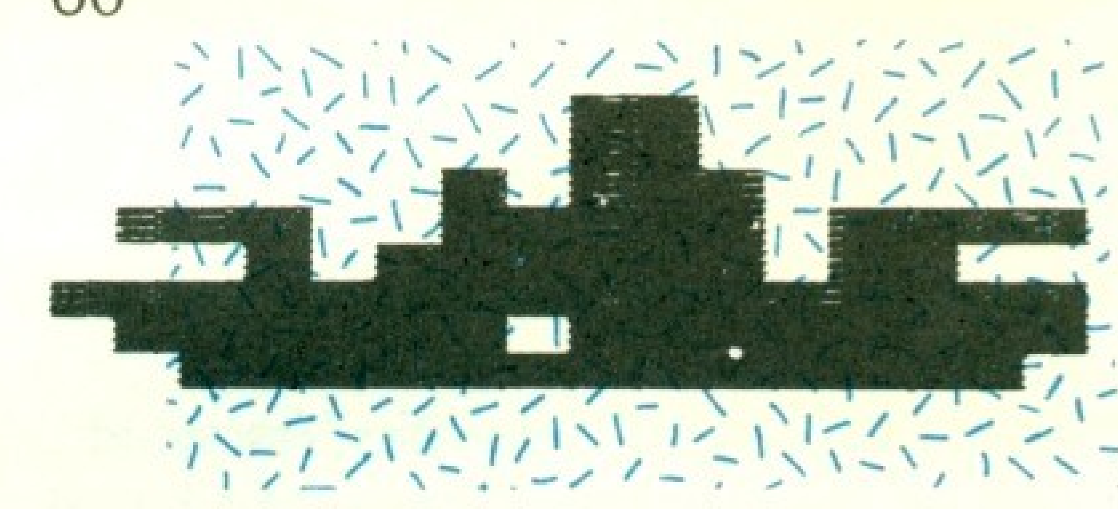
DATA 0,66,189,239,239,189,66,0
DATA 54,28,8,28,62,62,28,8

Tampoco nos olvidamos de los que se deleitan con las carreras automovilísticas, aquí les presentamos un hermoso fórmula uno, de cuatro caracteres, dos superiores y dos inferiores. Comenzamos con los primeros:

Izquierda: DATA 31,31,4,5,55,59,55,6
Derecha: DATA 248,248,32,160,236,220,236,96
(Parte inferior)
Izquierda: DATA 6,6,7,19,31,19,1,7
Derecha: DATA 96,96,224,200,248,200,128,224
Izquierda: DATA 224,159,119,239,239,119,159,224
Derecha: DATA 28,242,190,191,191,190,242,28
DATA 0,66,189,239,239,189,66,0
DATA 102,153,239,223,223,239,153,102

Y como no podía faltar, presentamos al señor PACMAN junto a sus alimentos preferidos.

Parámetro derecho:
DATA 60,127,252,240,240,252,127,60



Pacman, arriba:
DATA 66,66,231,231,255,255,126,60

Parámetro izquierdo:
DATA 60,254,63,15,15,63,254,60

Pacman, abajo:
DATA 60,126,255,231,231,66,60

Pacman con la boca cerrada:
DATA 60,126,255,255,255,255,126,60

Fantasma:
DATA 56,124,214,214,254,254,170,170

Por último veremos las frutas con que se alimenta este animalejo.

Frutilla:
DATA 24,82,247,255,255,126,60,24

Frambuesa:
DATA 4,8,8,86,171,213,106,60

Cereza:
DATA 8,8,20,20,34,99,243,96

Durazno:
DATA 44,110,231,247,247,247,102,44

Banana:
DATA 2,3,7,14,30,124,248,0

Ciruela:
DATA 8.16,24,60,124,62,60,24

Manzana:
DATA 24,82,255,255,255,255,126,36

Esperamos que el desfile de estrellas que acabamos de presenciar, les haya atomizado las neuronas. ¡A programar se ha dicho!

CAJA DE AHORROS

COMP.: TS 2068/SPECTRUM/TK90X
CLAS.: COMERCIAL
AUTOR: Fabio Tesolín



El siguiente programa tiene por finalidad calcular los intereses devengados según los movimientos realizados en nuestra caja de ahorros, pudiendo realizar un máximo de 5 extracciones en el mes.

Estructura del programa

30 - 70 Presentación
75 - 110 Ingreso de datos
800 - 2000 Movimientos del mes
3000 - 3040 Cálculo de intereses
3500 Rutina de sonido
4000 - 4499 Saldo negativo
5000 - 5020 Rutina de gráficos
5500 - 5510 Rutina de error de extracciones
5800 - 6000 Impresión de intereses

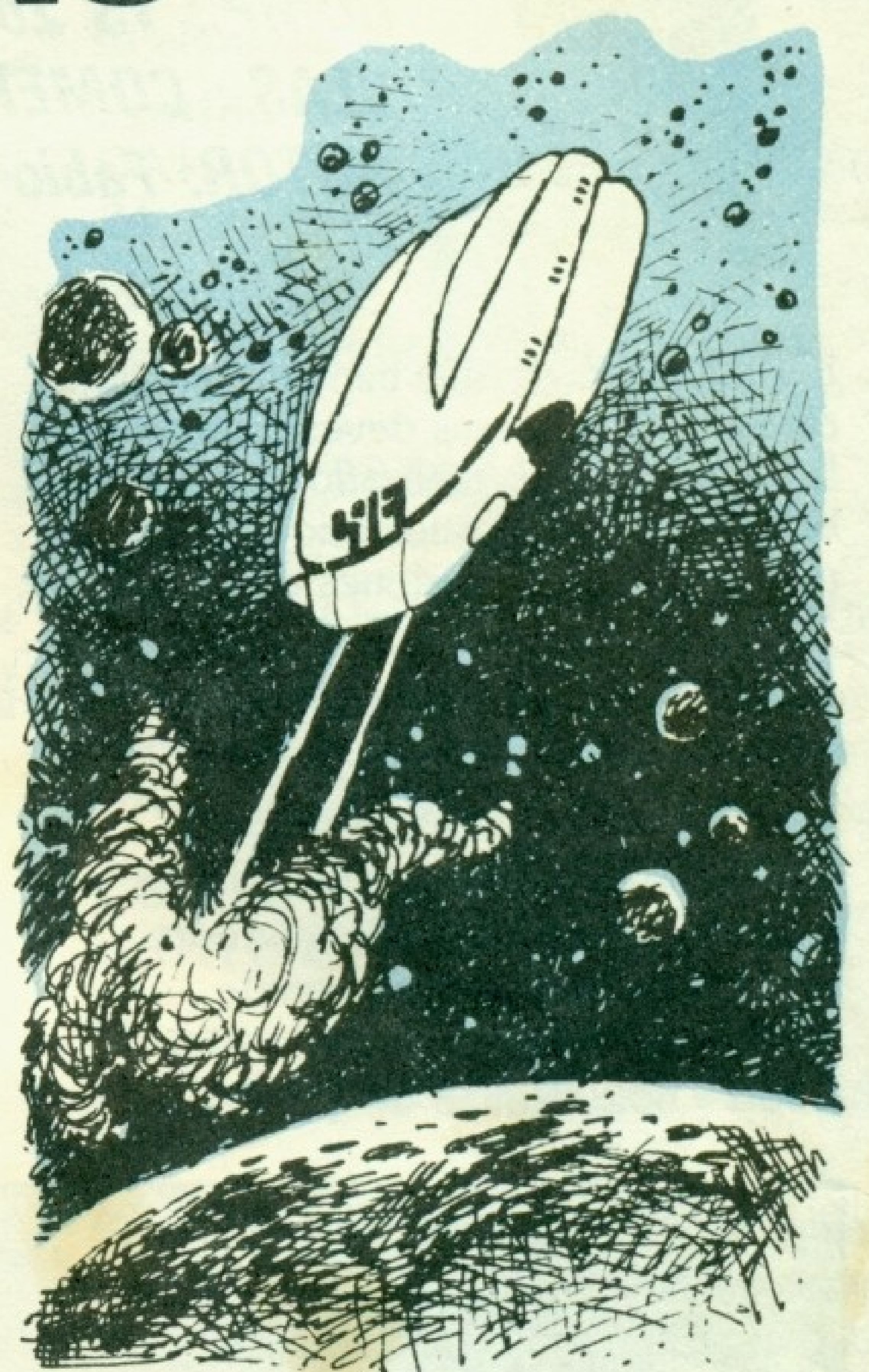
Variables útiles

A$: Nombre del titular
TA: Valor de la tasa anual
B$: Nombre del mes
MES: Cantidad de días del mes en curso
D(): Número de día en curso
I(): Importe correspondiente al día
INT: Intereses
S(): Monto

```
  30 CLEAR
  31 POKE 23609,30
  32 BORDER 6
  40 FLASH 1: PRINT AT 2,6;"* CA
JA DE AHORROS *": FLASH 0
  43 REM ???? PRESENTACION ?????
  45 PRINT AT 5,0;"Este programa
 calcula los inte- reses devenga
dos en el mes segunlos movimient
os realizados.     Tenga present
e que no podra ha- cer mas de ci
nco extracciones enel mes debido
 a que se trata de una Caja de A
horros Comun.
                    Al ingresar e
l mes  debera colo-car la primer
 letra en mayusculay a los impor
tes de extraccionesdebera antepo
ner el signo menos."
  50 GO SUB 3500: PRINT PAPER 6;
AT 21,0;"Pulse una tecla": PAUSE
 0
  60 CLS
  70 CLEAR : FLASH 1: PRINT AT 4
,6;"* CAJA DE AHORROS *": FLASH
0
  73 REM ???? TOMA DE DATOS ????
  75 INPUT "Titular ? ";a$: PRIN
T AT 8,2;"Titular:      ";a$: GO
SUB 3500
  80 INPUT "Tasa anual ? ";ta: P
RINT AT 12,2;"Tasa anual : ";ta;
" %": GO SUB 3500
  85 INPUT "Mes ? ";b$
  86 LET mes=0
  87 IF b$="Enero" OR b$="Marzo"
 OR b$="Mayo" OR b$="Julio" OR b
$="Agosto" OR b$="Octubre" OR b$
="Diciembre" THEN LET mes=31
  89 IF b$="Abril" OR b$="Junio"
 OR b$="Setiembre" OR b$="Septie
mbre" OR b$="Noviembre" THEN LET
 mes =30
  91 IF b$="Febrero bisiesto" TH
EN LET mes=29
  92 IF b$="Febrero" THEN LET me
s=28
  94 IF mes=0 THEN BEEP .8,-15:
GO 10 85
  96 PRINT AT 16,2;"Mes :
 ";b$: GO SUB 3500
  98 PRINT PAPER 6;AT 21,0;"Puls
e una tecla": PAUSE 0
 100 CLS
 110 GO SUB 5000
 800 REM ???? MOVIMIENTOS ?????
 900 DIM d(31)
 920 DIM i(31)
 930 DIM s(31)
 940 LET c=0
1000 FOR n=1 TO mes
1050 INPUT "Dia ? ";d(n)
1051 IF d(n)<0 OR d(n)>mes THEN
BEEP 0.8,-15: GO TO 1050
1052 IF d(n)=0 THEN GO TO 3000
1055 IF n=17 THEN COPY : CLS : G
O SUB 5000
1060 IF n<17 THEN PRINT AT n+3,1
;d(n): GO SUB 3500
1070 IF n>=17 THEN PRINT AT n-13
,1;d(n): GO SUB 3500
1200 INPUT "Importe ? ";i(n)
1202 IF i(n)<0 THEN LET c=c+1
1205 IF c=6 THEN GO TO 5500
1210 IF i(n)>0 AND n<17 THEN PRI
NT AT n+3,6;i(n): GO SUB 3500
1220 IF i(n)<0 AND n<17 THEN PRI
NT AT n+3,15;i(n): GO SUB 3500
1240 IF i(n)>0 AND n>=17 THEN PR
INT AT n-13,6;i(n): GO SUB 3500
1260 IF i(n)<0 AND n>=17 THEN PR
INT AT n-13,15;i(n): GO SUB 3500
1300 GO SUB 4500
1310 IF s(n)>0 AND n<17 THEN PRI
NT AT n+3,25;s(n)
1315 IF s(n)>0 AND n>=17 THEN PR
INT AT n-13,25;s(n)
1320 IF s(n)<0 THEN GO TO 4000
2000 NEXT n
2980 REM ????? INTERESES ?????
3000 LET int=0
3005 FOR n=1 TO mes
3010 IF d(n+1)=0 THEN LET d(n+1)
=mes
3020 LET int=int+(s(n)*(d(n+1)-d
(n))*ta)/36500
3030 IF d(n+1)=mes THEN GO TO 58
00
3040 NEXT n
3500 BEEP .05,15: BEEP .05,15: B
EEP .05,15: RETURN
3980 REM ???? SALDO ROJO ????
4000 FLASH 1: PRINT PAPER 2; INK
 7;AT 10,4;"**    SALDO EN ROJO
 **";AT 11,4;"      NO COMPATIBLE
     ";AT 12,4;"CON CRITERIOS CO
NTABLES": FLASH 0
4010 BEEP .8,-15: COPY : PAUSE 1
000: CLEAR : GO TO 70
4499 GO TO 6000
4500 LET s(n)=i(1)+i(2)+i(3)+i(4
)+i(5)+i(6)+i(7)+i(8)+i(9)+i(10)
+i(11)+i(12)+i(13)+i(14)+i(15)+i
(16)+i(17)+i(18)+i(19)+i(20)+i(2
1)+i(22)+i(23)+i(24)+i(25)+i(26)
+i(27)+i(28)+i(29)+i(30)+i(31)
4502 RETURN
4980 REM ???? GRAFICO ????
5000 PRINT AT 0,0;"          CAJA
DE AHORROS           ";AT 1,0;" TI
TULAR: ";a$;AT 2,0;" DIA DEPOSIT
O  EXTRACC.  SALDO  "
5010 DRAW 0,175: PLOT 255,0: DRA
W 0,175: PLOT 0,0: DRAW 255,0: P
LOT 40,0: DRAW 0,155: PLOT 112,0
: DRAW 0,155: PLOT 184,0: DRAW 0
,155
5020 RETURN
5490 REM ???? ERROR EXTRAC. ???
5500 FLASH 1: PRINT PAPER 1; INK
 7;AT 10,4;"OPERACION NO ADMITID
A";AT 11,4;"MAXIMO 5 EXTRACCIONE
S": FLASH 0
5510 BEEP .8,-15: COPY : PAUSE 1
000: CLEAR : GO TO 70
5790 REM ??? IMPRIMIR INTERES???
5800 FLASH 1: PRINT PAPER 3; INK
 7;AT 20,3;"Interes :  A ";int;"
 .-": FLASH 0
5805 FOR h=-30 TO 30 STEP 10: BE
EP .08,h: NEXT h
5810 COPY : PAUSE 1000: CLEAR :
GO TO 70
6000 COPY : PAUSE 1000: CLS : GO
 TO 70
```

PROGRAMAS DESTRUCTOR DE ESTRELLAS

COMP.: CZ 1000/1500 TK 83/85
CONF.: 1 K
CLAS.: JUEGO



Una vez tipeado y ejecutada la instrucción RUN, el programa nos preguntará por el Skill (nivel de dificultad) que puede variar de uno a 29.
La nave espacial es posicionada en una de las muchas columnas de la pantalla. Las estrellas aparecen a la derecha de ésta, moviéndose a la izquierda. Usted puede dispararles con la tecla '6' hacia abajo y con la tecla '7' hacia arriba.
Los caracteres gráficos de la línea 40 se obtienen con las teclas 'Q', 'G' y 'W' en modo gráfico.
El de la línea 100 es un ':' invertido.

Variables útiles

S: contador de la fila de la bala.
A: número asociado a la columna de la nave, variado según el nivel de dificultad.
B: número de fila de la estrella (aleatorio).
C: contador de la columna de la estrella.
M: contador de intentos.

Estructura del programa

5-30: Entrada del nivel de dificultad
35-130: Lógica simple del programa.

```
  5 LET S=0
 10 PRINT "SKILL"
 20 INPUT A
 30 CLS
 35 FOR M=1 TO 10
 40 PRINT AT 0,A+2;"▄█▄"
 50 LET B=INT (RND*15)+1
 60 FOR C=0 TO 30
 70 PRINT AT B,C;" *"
 80 PRINT AT S,A+3;" "
 90 LET S=S+(INKEY$="7")-(INKEY
$="6")
100 PRINT AT S,A+3;"█"
110 NEXT C
120 NEXT M
130 IF M>10 THEN PRINT "FIN"
```

MUSIC MACHINE

COMP.: CZ 1000/1500; TK 83/85
CONF.: 1 K
CLAS.: UTILITARIO



Este programa es el más frívolo de los utilitarios que jamás se han publicado.
Usa los comandos FAST y SLOW para generar una benévola escala musical, a través del parlante del televisor.
El sonido logrado es el bajo profundo asociado generalmente, con el que fluye de la cripta del Conde Drácula, cuando toca el "Rock de la media noche" en su órgano.
La escala es generada por las teclas del 1 al 6.
Sintonice el dial suavemente hasta escuchar ese tenebroso susurro, suba el volumen y sacuda su espíritu al son de un Vampiro Romantic Rock.

```
  1 REM MUSIC MACHINE
  5 FAST
  6 SLOW
 25 IF INKEY$="3" THEN GOTO COD
E "C"
 27 IF INKEY$="4" THEN GOTO VAL
"80"
 28 IF INKEY$="5" THEN GOTO COD
E "▒"
 29 IF INKEY$="6" THEN GOTO VAL
"200"
 30 GOTO CODE "▪"
 40 FAST
 41 SLOW
 50 SLOW
 60 FAST
 65 IF INKEY$="2" THEN GOTO COD
E "▌"
 67 IF INKEY$="4" THEN GOTO VAL
"80"
 68 IF INKEY$="5" THEN GOTO COD
E "▒"
 69 IF INKEY$="6" THEN GOTO VAL
"200"
 70 GOTO CODE "C"
 80 FAST
 81 FAST
 83 SLOW
 90 SLOW
 91 SLOW
110 FAST
120 IF INKEY$="2" THEN GOTO COD
E "▌"
125 IF INKEY$="3" THEN GOTO COD
E "C"
126 IF INKEY$="5" THEN GOTO COD
E "▒"
127 IF INKEY$="6" THEN GOTO VAL
"200"
130 GOTO VAL "80"
140 FAST
150 SLOW
152 SLOW
153 SLOW
154 SLOW
160 FAST
170 IF INKEY$="4" THEN GOTO VAL
"80"
171 IF INKEY$="2" THEN GOTO COD
E "▌"
172 IF INKEY$="3" THEN GOTO COD
E "C"
173 IF INKEY$="6" THEN GOTO VAL
"200"
180 GOTO CODE "▒"
200 FAST
210 FAST
220 SLOW
221 SLOW
222 SLOW
223 SLOW
229 SLOW
230 FAST
240 IF INKEY$="2" THEN GOTO COD
E "▌"
241 IF INKEY$="3" THEN GOTO COD
E "C"
242 IF INKEY$="4" THEN GOTO VAL
"80"
243 IF INKEY$="5" THEN GOTO COD
E "▒"
250 GOTO VAL "200"
```

PROGRAMAS FACTORES

COMP.: CZ 1000/1500; TK 83/85
CONF.: 1 K
CLAS.: EDUCATIVO

8={2,2,2}

4={2,2}

12={3,2,2}

14={7,2}

18={3,3,2}

20={5,2,2}

21={7,3}

Estos son en realidad dos programas destinados al cálculo de los factores que componen un número determinado. Ambos son de seguimiento simple y fácil manejo. El primero y más corto de ellos cumple con la tarea de calcular todos los posibles factores del número consultado. El segundo tiene como misión calcular solamente los factores primos que componen a dicho número.

```
  5 REM FACTORES PRIMOS
 20 INPUT A
 30 PRINT A;"■";
 40 LET B=2
 50 IF A/B<>INT (A/B) THEN GOTO
80
 60 GOSUB 190
 70 GOTO 50
 80 FOR B=3 TO A STEP 2
 90 FOR C=3 TO B-1 STEP 2
100 IF B/C=INT (B/C) THEN GOTO
150
110 NEXT C
120 IF A/B<>INT (A/B) THEN GOTO
150
130 GOSUB 190
140 GOTO 120
150 NEXT B
160 POKE 16398,(PEEK 16398)-1
170 PRINT "."
180 RUN
190 PRINT B;"*";
200 LET A=A/B
210 RETURN
```

Si por los avatares del destino, o por sufrir alguna de las diez encarnaciones de Vichnu' (si saben quién es por favor infórmennos), se ha olvidado de lo que es un número primo, le recordamos que son los que cumplen ser divisibles solamente por sí mismos y por la unidad. El número 1, aunque cumple con el legado anterior, queda excluido de esta familia de números.

Ejemplos: 2, 3, 5, 7, 11, 13...

```
 5 REM ************************
 6 REM *                      *
 7 REM *  CALCULO DE TODOS    *
 8 REM *  LOS FACTORES DE     *
 9 REM *      UN NUMERO       *
10 REM ************************
11 FAST
20 INPUT A
30 PRINT A;
40 FOR B=A-1 TO 1 STEP -1
50 IF A/B=INT (A/B) THEN PRINT
,";B;
60 NEXT B
70 PRINT "."
80 RUN
```

Si

INTERES COMPUESTO

COMP.: CZ 1000/1500 TK 83/85
CONF.: 1 K
CLAS.: COMERCIAL

Si detestamos las cuentas que llevan calcular el interés compuesto, este programa será para nosotros.

Hará trabajar a la computadora automáticamente bajo las reglas del interés compuesto, una vez que le hayamos informado sobre el monto inicial, la razón del interés, y el número de años a considerar. Está también preparado para resolver problemas de interés simple.

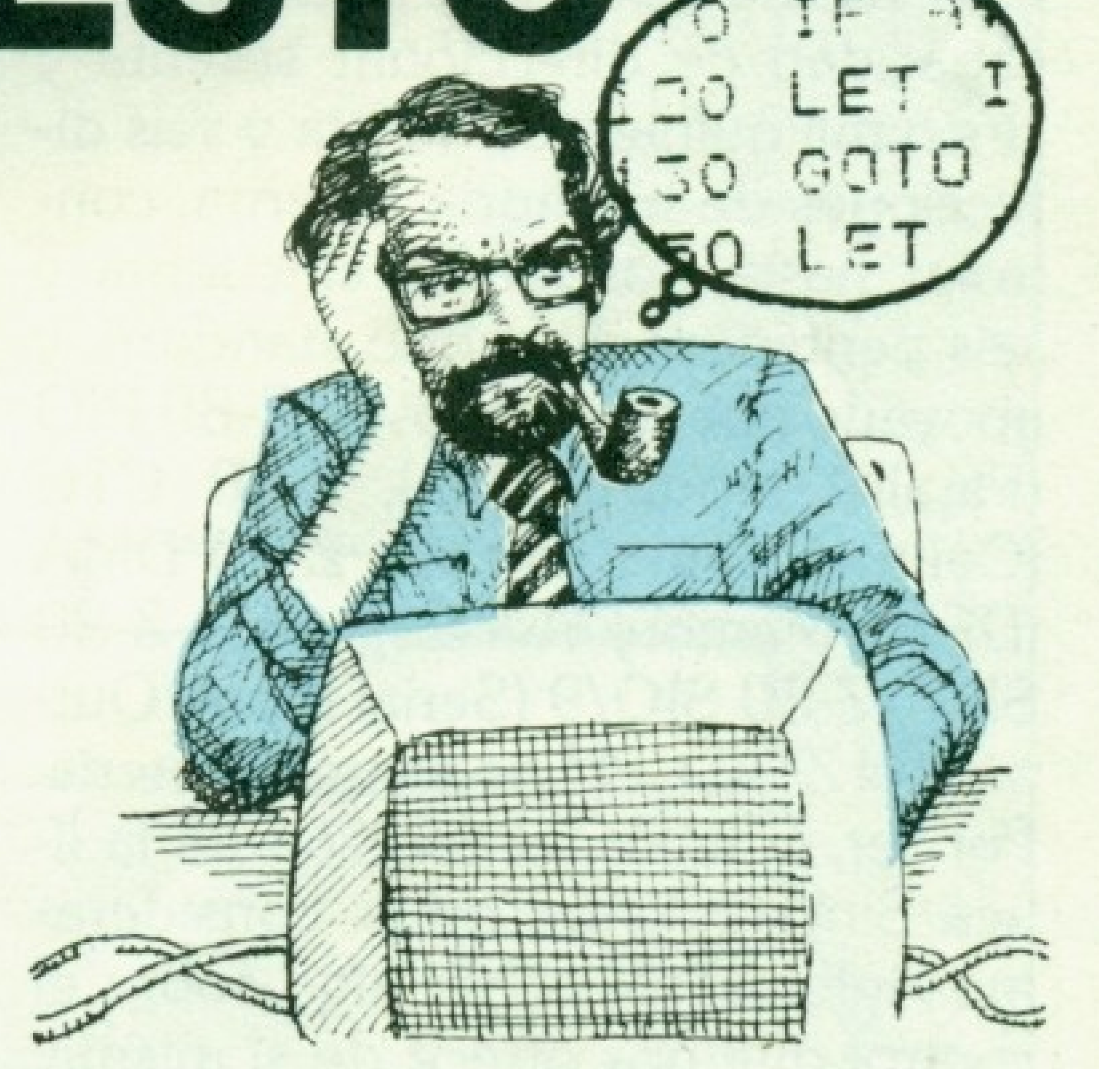
Es un programa que será de gran ayuda en el trabajo hogareño.

....

5: CLS
10: PRINT "MONTO INICIAL:"
20: INPUT P
30: PRINT "TIEMPO:"
40: INPUT T
50: PRINT "RAZON ANUAL:"
60: INPUT R
70: PRINT
80: PRINT "ENTRE ""IS"" PARA INTERES SIMPLE"
90: PRINT "ENTRE ""IC"" PARA INTERES COMPUESTO"
100: INPUT A$
105: CLS
110: IF A$ ="IS" THEN GOTO 150
120: LET I =P*((1 +R/100)**T) −P
130: GOTO 200
150: LET I =(P*R*T)/100
200: PRINT "INTERES =A";I
202: PRINT
205: PRINT "TOTAL =A";I +P
206: PRINT
210: PRINT "ESO ES TODO? (S/N)"
220: INPUT B$
230: IF B$ ="N" THEN RUN

Variables importantes

P: Monto inicial.
T: Tiempo en años.
R: Razón anual.
A$: Variable de control.
B$: Idem A$.
I: Interés calculado.

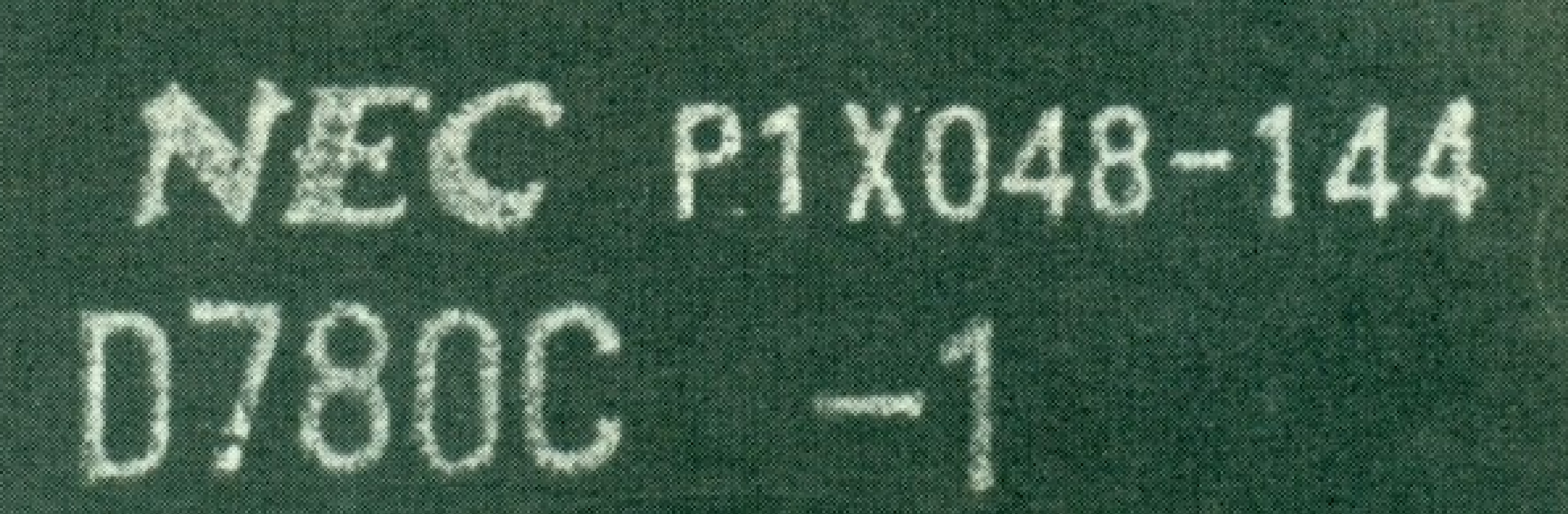


Estructura del programa

5-70: Entrada de datos descriptivos.
80-110: Opción de un tipo de interés.
120: Cálculo del interés compuesto.
150: Cálculo del interés simple.
200-230: Obtención de resultados y opción de nuevo cálculo.

Z-80 CPU Z-80A CPU

Examinaremos el significado y la utilización de cada uno de los pines de la unidad central de proceso Z-80, que es el corazón de las máquinas Sinclair, CZ, TK y compatibles. Esto facilitará la comprensión y seguimiento de circuitos electrónicos cuya lógica digital se base en este microprocesador.



Dentro del mundillo de las unidades centrales de proceso, con ocho bits de datos, y dieciseis bits de direcciones, presentamos al Rey de los procesadores.

Con más de seiscientas instrucciones, capacidad de direccionar sesenta y cinco mil quinientos treinta y seis direcciones de memoria externa, controlar hasta doscientos cincuenta y seis periféricos de modo standard, y apoyado por sus primos, el Z-80 PIO (Parallel Input/Output), Z-80 CTC (Counter Timer Circuit), Z-80 DMA (Direct Memory Access), y los Z-80 SIO y Z-80 SIO/9 (Serial Input/Output); el Z-80 CPU no es una tontería. Por ser, además, el corazón de la línea Sinclair, entre otras, consideramos interesante saber algo sobre el control que nos ofrece de sí misma, esta cucaracha de cuarenta patas.

Justamente, apoyándonos en la figura, veremos una lista que nos aclarará el significado y utilización de cada una de esas patas.

FIGURA

Comencemos con la listita entonces.

Nombre: A0-A15
Pines: 1 a 5, y 30 a 40.
Descripción: bus de direcciones de 16 bits.

Nombre: D0-D7
Pines: 7 a 10, y 12 a 15.
Descripción: bus de datos bidireccional de 8 bits.

Nombre: 0/CLOCK
Pin: 6
Descripción: entrada de pulsos del reloj que controlará la velocidad de la CPU.

Nombre: Vcc
Pin: 11
Descripción: alimentación del chip, su valor es de +5 volts.

Nombre: INT (negado)
Pin: 16
Descripción: (Interrupt Request). Cuando a esta línea se le manda un bajo (0 volt), comienza un ciclo de interrupción siempre que el indicador de interrupciones (flag IFF) haya sido seteado, por software. De otra forma esta línea es ignorada.

La razón de que el nombre figure negado, es que la línea al igual que otras, necesita un "bajo" para ser activada.

Nombre: NMI (negado)
Pin: 17
Descripción: (Nonmaskable Interrupt). Cuando a esta línea se le entrega un nivel bajo de tensión, provoca que el Z-80 detenga el programa que está ejecutando y efectúe un Call o llamada a la posición de memoria 66h (hexa). Este tipo de interrupción no enmascarable, no es posible de evitar por software.

Nombre: HALT (negado)
Pin: 18
Descripción: por medio de esta señal el Z-80 indica que ha ejecutado la instrucción Halt, esperando que se produzca alguna de las dos interrupciones anteriores, antes de que puedan reanudarse las operaciones.

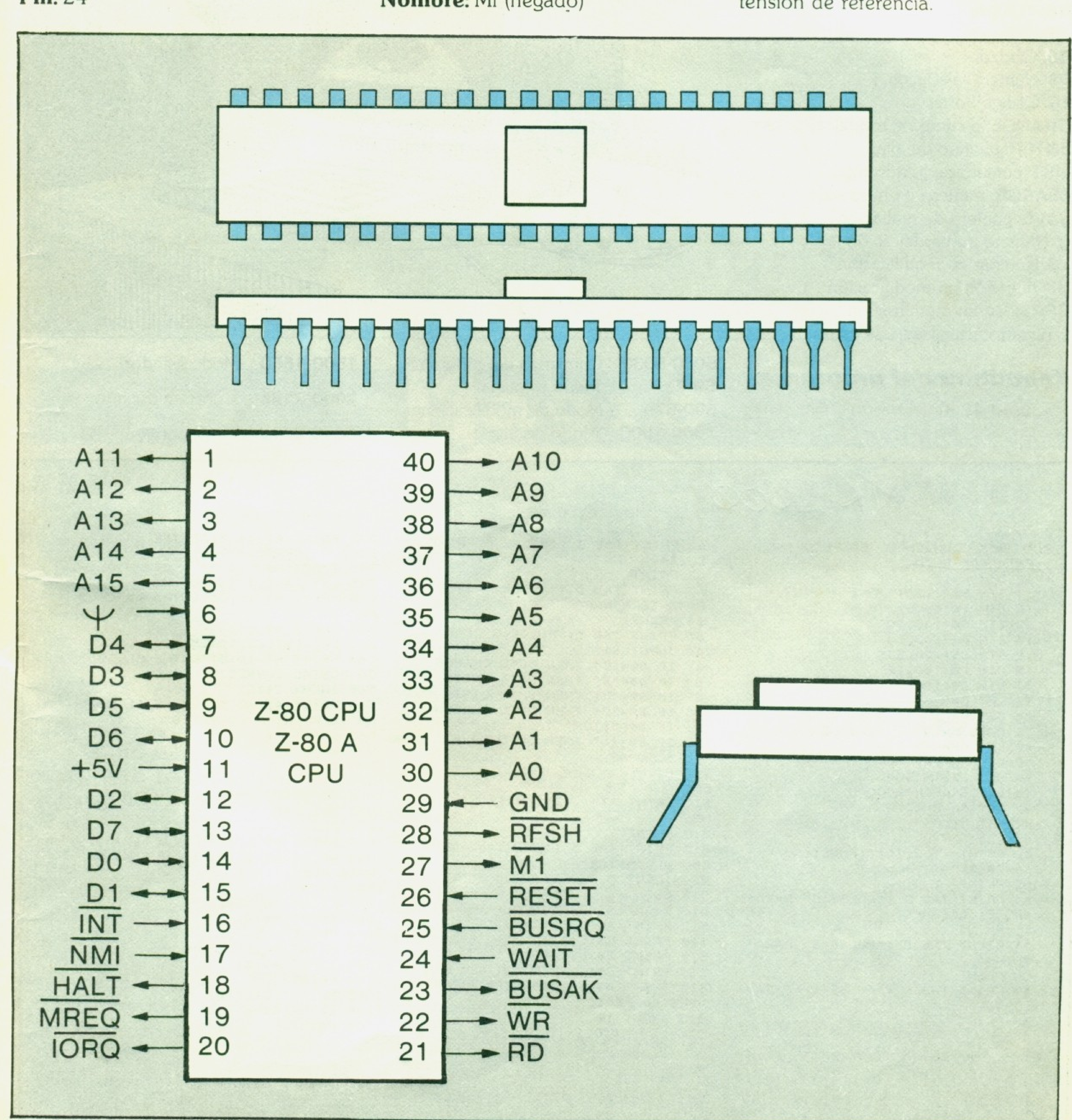
Nombre: MREQ (negado)
Pin: 19
Descripción: (Memory Request). Esta línea manda un "bajo" cuando el Z-80 está listo para acceder a una posición de memoria.

Nombre: IORQ (negado)
Pin: 20
Descripción: (In/Out Request). esta línea manda un "bajo" cuando el Z-80 está listo para acceder a un pórtico de Entrada/Salida.

Nombre: RD (negado)
Pin: 21
Descripción: (Read). Esta línea manda un bajo cuando el Z-80 necesita leer un dato de una posición de memoria o de un pórtico.

Nombre: WR (negado)
Pin: 22
Descripción: (Write). Esta línea

manda un bajo cuando el Z-80 necesita "escribir" un dato en una posición de memoria o en una división de Entrada/Salida.
Nombre: BUSAK (negado)
Pin: 23
Descripción: (Bus Acknowledge). Esta línea es usada para indicar qué datos, direcciones y señales de control de la CPU han sido desactivados, para que una división externa pueda ahora controlar estas señales.
Nombre: WAIT (negado)
Pin: 24
Descripción: Al poner un bajo en este pin el Z-80 entrará en un ciclo de espera, mientras dure este pulso.
Nombre: BUSRQ (negado)
Pin: 25
Descripción: (Bus Request). Esta línea es usada para indicarle al Z-80 que un periférico tomará el control del bus de datos y de direcciones.
Nombre: RESET (negado)
Pin: 26
Descripción: Un pulso bajo en este pin causa la inicialización de la CPU.
Nombre: Mi (negado)
Pin: 27
Descripción: Esta señal se usa para indicar que la CPU realiza el ciclo que produce la instrucción Mi.
Nombre: RFSH (negado)
Pin: 28
Descripción: (Refresh). Señal que manda el Z-80 para mantener vivos los datos en el bus de direcciones (RAMs dinámicas).
Nombre: GND
Pin: 29
Descripción: Masa o nivel bajo de tensión de referencia.





COMP.: CZ 1000/1500; TK 83/85
CONF.: 16 K
CLAS.: COMERCIAL

Lista de variables:

S$: Nombre a buscar
T$: Ciudad a buscar
N$: Nombre
A$: Dirección
C$: Ciudad
P$: Número telefónico
Z$: Código Postal
CHANGE: puntero de modificaciones
ENTER: puntero de altas
LIST: puntero de listado total
SEARCH: puntero de búsqueda
SAVE: puntero de grabación
C: Número de modificación
Y$: Nombre en modificación
H$: dirección en modificación
G$: ciudad en modificación
L: número actualizado de fichas

Estructura del programa

10-30/40-49 Inicialización de variables
5000-5030 Comienzo del programa
50-75 Menú principal
500-570 Modo de modificaciones
1000-1500 Modo de altas
1500-1550 Modo listador
2000-2166 Modo de búsqueda
6000-6010 Modo de grabación

```
1  REM  COMIENCE  EL PROGRAMA
ENTRANDO GOTO 35
10 DIM L(1)
11 DIM S$(1,30)
12 DIM T$(1,30)
13 DIM N$(100,30)
14 DIM A$(100,30)
15 DIM C$(100,25)
16 DIM P$(100,12)
17 DIM Z$(100,7)
25 LET L=0
30 LET N=0
35 GOTO 5000
40 LET CHANGE=500
42 LET ENTER=1000
44 LET LIST=1500
46 LET SEARCH=2000
47 LET SAVE=6000
48 LET N=L
49 CLS
50 PRINT AT 2,10;":FUNCION:"
51 FAST
52 PRINT
54 PRINT TAB 5 ;"PULSE C  PARA
MODIFICACIONES"
55 PRINT
56 PRINT TAB 5 ;"PULSE E  PARA
ALTAS"
57 PRINT
59 PRINT TAB 5 ;"PULSE L  PARA
LISTAR"
60 PRINT
61 PRINT TAB 5 ;"PULSE S  PARA
BUSCAR"
62 PRINT
63 PRINT TAB 5 ;"PULSE ""STOP""
PARA TERMINAR"
64 PRINT
65 PRINT TAB 5;"PULSE X  PARA
66 INPUT B$
67 IF B$="C" THEN GOTO CHANGE
68 IF B$="B" THEN GOTO ENTER
69 IF B$="L" THEN GOTO LIST
70 IF B$="S" THEN GOTO SEARCH
71 IF B$="STOP" THEN STOP
72 IF B$="X" THEN GOTO SAVE
75 GOTO 49
113 PRINT
500 CLS
510 PRINT AT 1,12;"MODIFICACIO
NES"
512 PRINT
513 PRINT TAB 5 ;"ENTRE EL NUME
ERO A CAMBIAR"
514 INPUT C
515 CLS
516 PRINT AT 2,0;N$(C,1 TO 30)
517 PRINT A$(C,1 TO 30)
518 PRINT C$(C,1 TO 25)
519 PRINT Z$(C,1 TO 7)
520 PRINT P$(C,1 TO 12)
524 PRINT AT 10,5;"ENTRE EL NOM
MBRE CORRECTO"
525 INPUT I$
530 LET N$(C,1 TO 30)=I$
535 PRINT AT 10,5;"ENTRE LA DIRE
ECCION CORRECTA"
540 INPUT H$
545 LET A$(C,1 TO 30)=H$
550 PRINT AT 10,5;"ENTRE LA CIUD
AD"
555 INPUT G$
560 LET C$(C,1 TO 25)=G$
562 PRINT AT 10,5;"ENTRE EL CODI
GO POSTAL CORRECTO"
563 INPUT Z$(C)
564 PRINT AT 10,5;"ENTRE EL NUME
RO TELEFONICO CORRECTO"
565 INPUT P$(C)
570 GOTO 49
1000 CLS
1010 FOR X=N+1 TO 100
1015 IF X=100 THEN GOTO 1142
1020 LET L=X
1030 CLS
1040 PRINT AT 1,10;"   ALTAS    "
1050 PRINT AT 2,10;"LA ULTIMA EN
TRADA FUE: ";X-1
1052 PRINT
1055 PRINT "NOMBRE: "
1060 INPUT N$(X)
1070 PRINT
1075 PRINT "DIRECCION: "
1080 INPUT A$(X)
1090 PRINT
1095 PRINT "CIUDAD: "
1100 INPUT C$(X)
1105 PRINT
```

```
1107 PRINT "CODIGO POSTAL: "
1108 INPUT Z$(X)
1109 PRINT
1110 PRINT "NUMERO TELEFONICO: "
1111 INPUT P$(X)
1112 PRINT
1115 PRINT " OTRA ENTRADA ? (S/
N)"
1130 INPUT F$
1138 IF F$ <> "S" THEN GOTO 47
1140 NEXT X
1142 PRINT
1145 PRINT "              ARCHIVO CO
MPLETO"
1147 PAUSE 200
1150 GOTO 47
1500 CLS
1505 PRINT AT 20,12;" LISTADO "
1510 FOR V=1 TO L
1515 SCROLL
1520 PRINT N$(V);V
1521 SCROLL
1522 PRINT A$(V)
1523 SCROLL
1524 PRINT C$(V)
1525 SCROLL
1526 PRINT Z$(V)
1527 SCROLL
1528 PRINT P$(V)
1529 SCROLL
1530 PRINT
1531 PAUSE 300
1550 GOTO 49
2000 CLS
2020 PRINT AT 1,12;"MODO DE BUSQ
UEDA"
2021 PRINT
2022 PRINT "BUSCA NOMBRE (N) O C
IUDAD (C) ?"
2023 INPUT V$
2034 IF V$="N" THEN GOTO 2050
2036 PRINT "ENTRE EL NOMBRE DE L
A CIUDAD"
2037 PRINT "POR FAVOR TIPEE CORR
ECTAMENTE"
2038 INPUT T$(1,1 TO 25)
2039 FOR S=1 TO L
2040 IF C$(S,1 TO 25)=T$(1,1 TO
25) THEN GOTO 2160
2041 NEXT S
2042 PRINT
2043 PRINT TAB 5 ;"        FIN DE
 LISTA"
2044 PAUSE 3000
2045 GOTO 47
2047 PRINT
2050 PRINT "ENTRE EL NOMBRE A BU
SCAR: "
2055 PRINT
2060 INPUT S$(1,1 TO 30)
2063 FOR S=1 TO L
2065 IF N$(S,1 TO 30)=S$(1,1 TO
30) THEN GOTO 2160
2070 NEXT S
2100 PRINT
2110 PRINT "NOMBRE INEXISTENTE"
2115 PAUSE 3000
2120 GOTO 47
2140 PRINT N$(S)
2141 PRINT A$(S)
2142 PRINT C$(S)
2143 PRINT Z$(S)
2144 PRINT P$(S)
2145 GOTO 47
2146 PAUSE 3000
2160 PRINT N$(S)
2161 PRINT A$(S)
2162 PRINT C$(S)
2163 PRINT Z$(S)
2164 PRINT P$(S)
2165 PRINT
2166 GOTO 2041
5000 PRINT AT 2,0;"************
******************"
5005 PRINT "*
                             *"
5006 PRINT "*
                             *"
5007 PRINT "*
                             *"
5008 PRINT "*
                             *"
5009 PRINT "*
                             *"
5010 PRINT "*
                             *"
5011 PRINT "*
                             *"
5012 PRINT "******************
************"
5015 PRINT 6,5;"ARCHIVO TELEFONI
CO"
5020 PAUSE 300
5021 CLS
5022 PRINT AT 4,0;"ESTE PROGRAMA
 PUEDE ARCIVAR"
5023 PRINT
5024 PRINT "100 NOMBRES,DIRECCIO
NES Y TE"
5025 PRINT
5026 PRINT "TAMBIEN PUEDE BUSCAR
POR NOM-"
5027 PRINT
5028 PRINT "BRE O CIUDAD."
5029 PAUSE 500
5030 GOTO 38
6000 SAVE "ARCHI-T.E."
6010 GOTO 38
7000 SAVE "ARCHI-T.E."
7010 RUN
```

Magic

UNA LINEA 0:

La siguiente instrucción causará que la línea editada en el programa con el número 1, pase a ser la línea 0, que no podrá editarse luego ni cambiar su contenido.
POKE (PEEK 23635 +256*PEEK 23636 +1),0
Entremos la instrucción anterior en la Spectrum sin número de línea y veremos aparecer en el listado una hermosa linea 0 donde antes se encontraba la línea 1.
La forma de volver a tener control sobre esa línea es entrar nuevamente el POKE, pero esta vez con el número 1 después de la coma.
Este mismo efecto podrá lograrse en la 2068 por medio de la siguiente instrucción:
POKE 26711,0
Prueben de colocar números más grandes que uno o números de línea que ya existan, si está loca.

"LINE-ADDR":

Esta subrutina situada en la ROM de la 1000/1500, comienza en la posición 09D8 y sirve para averiguar en qué dirección empieza una determinada línea BASIC.
Una de las utilidades es la de localizar una determinada línea y luego borrar el bloque BASIC en el EDITOR. precisa como dato el número de línea en el registro HL (registro interno del Z-80), devolviendo la dirección de la línea en la misma pareja HL.

RANDOMIZE USR 3652:

Realiza un scroll de la mitad inferior de la pantalla incluido el borde.

RANDOMIZE USR 4317:

Cambia automáticamente de mayúsculas a minúsculas o viceversa según se encuentre en ese momento.

POKE 23617,236:

Si es colocado delante de un INPUT, en el lugar del cursor habitual saldrá un signo de interrogación.

RANDOMIZE USR 1331:

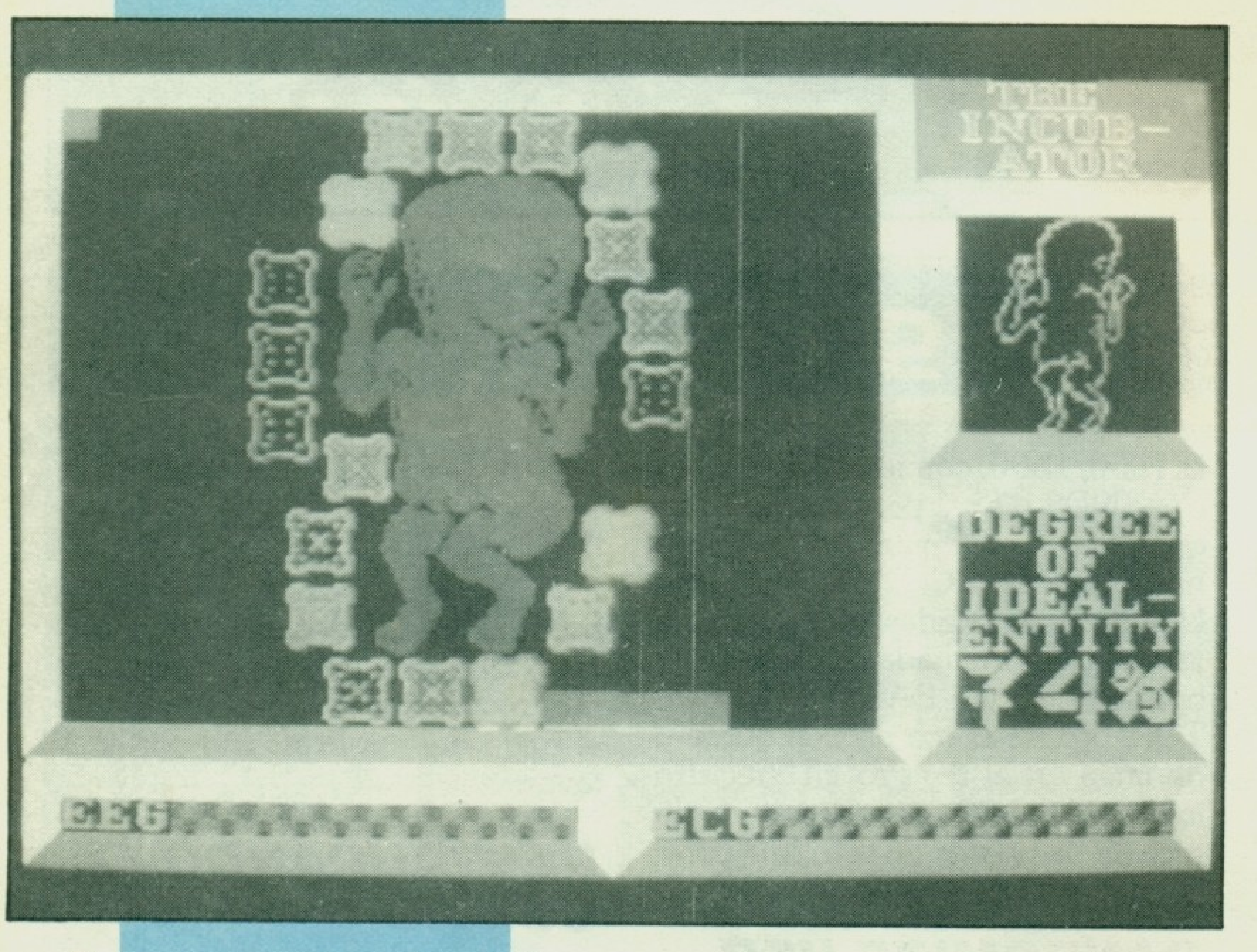
Esta llamada a la ROM, ejecuta en el borde de la pantalla unas rayas parecidas a las que vemos cuando cargamos un programa y además produce el ruido de una explosión por el altavoz.

RESTAURANDO EL COLOR:

Esta es una pequeña rutina assembler que provoca que los atributos de borde papel y tinta vuelvan a ser los mismos que al encender nuestra Spectrum de 48 ó 16 K.
10 DATA 62,56,1,141,92,2,1,72,2,201
20 CLEAR (USR "A"−12)
30 FLOR N=(USR "A"−11) TO (USR "A"−1)
40 READ A: POKE N,A: NEXT N
Para llamarla en el caso de la Spectrum de 48 K usaremos: RANDOMIZE USR 65357, y en el caso de ser de 16 K: RANDOMIZE USR 32589

GENESIS DE LA PROGRAMACION

Los guiamos en los primeros pasos para lograr un buen software. Para ello será indispensable saber como utilizar el cursor, entrar comandos o borrar caracteres (que pueden encontrar muy claro en el manual de la máquina).



Al comprar nuestro primer ordenador, con frenética ansiedad tratamos de devorarnos el manual "específico" que lo acompaña.

Probamos los ejemplos, los cambiamos, descubrimos con los ojos muy abiertos, algunas de las cosas que dice ese manual que nuestra computadora puede hacer. Y comienza el vicio...

Comenzamos a ahorrar memoria. acortamos los nombres de las variables hasta dejarlos de un caracter de largo, localizamos todas las subrutinas al comienzo del programa para ganar en tiempo de ejecución, florecen los GOTO, hasta hacer del seguimiento del programa un jeroglífico indescifrable, inclusive para nosotros mismos. Y cuando al mes de haber terminado nuestro programa, se nos ocurre que además debería hacer tal o cual cosa, y vemos el listado con DATAs por cualquier parte, o no encontramos el corazón de éste, notamos que un sentimiento de angustia y desesperación nos invade hasta las lágrimas. Sin duda algo anda mal.

Y es cierto, no somos para nada "estructurados" en la forma de programar. Obviamente no es nuestra culpa, en el manualcito de la máquina no decía nada sobre cómo ordenarnos, ni de cómo hacer un buen programa.

La llamada Programación Estructurada es la que se encarga de resolver este tipo de problemas.

Este tipo de programación tiene ciertas reglas que de cumplirlas (no exageradamente), harán que nuestros programas sean comprensibles, no sólo por nosotros sino por todos los que estén en el tema. A no desesperar que nadie les robará sus programas, siempre hay tiempo e instrucciones para complicar y proteger nuestras creaciones.

REGLAS PARA LOGRAR UN BUEN PROGRAMA

- Lo primero que hay que tener en cuenta, es todo lo que queremos que haga el programa.

Concentrarnos en los bloques de comienzo, parte central y fin, hasta llegar a plantearnos los posibles problemas, lo más preciso que podamos.

- Debemos estructurar cuidadosamente el programa por bloques, y asignarle a cada uno de éstos una tarea específica.
- Sin rigor, el programa debe tener un comentario que aclare el funcionamiento o finalidad en cada línea.
- La lógica de programación y su trayecto a lo largo del programa, han de ser fáciles de seguir. Sin duda una buena documentación de éste, en lo que respecta a los algoritmos utilizados, será de gran ayuda a la hora de la revisión del mismo.

Lo que aquí se definen como reglas de programación, aplicadas a sus proyectos, harán que el suyo sea un "buen" programa. Siguiendo esta línea de programación estructurada, notará una mayor facilidad de desenvolvimiento frente a los problemas planteados.

LINEALIDAD EN LA PROGRAMACION

Cuando se dice que un programa tiene que ser fácil de seguir, se está haciendo referencia directa a la programación lineal.

Esto es muy simple, un programa "comienza al principio y acaba al final", parece tonto, ¿pero se fijaron en la cantidad de programas hechos por profesionales en los que esto no ocu-

rre?, la gran mayoría.
La mejor forma de entender a fondo este concepto es justamente programando. Recordemos que el programa debe estar partido en varios bloques, sub-programas o subrutinas, que tienen que ser lo más cortas posibles.
Una subrutina (al igual que el programa), tiene un comienzo y un fin bien determinados, entre los cuales se da entrada a la información, se la procesa y se entrega el resultado acorde a este proceso, que en este caso se llaman punto de entrada y de salida, respectivamente.
Una subrutina puede perfectamente hacer referencia a otra, formando una cadena de subrutinas anidadas, siempre y cuando mantengamos el concepto de linealidad, y recordemos que cada una debe cumplir con su tarea específica.
Estas son puestas dentro del programa en el orden de llamada que efectuaran ellas mismas. Contribuyendo de este modo al seguimiento del mismo.
Existe una gran rivalidad entre los programadores que prefieren la programación del tipo pregunta/respuesta y los que la prefieren estructurada. Es nuestra intención tratar de lograr un punto de encuentro entre ambas, que haga versátil, creativa, cómoda y rápida, la implementación de programas. Sin lugar a dudas todos los que comenzamos a programar por la curiosidad que nos producían estos aparatitos, en forma casera, somos por lo menos programadores de tendencia pregunta/respuesta, porque además el BASIC es un lenguaje que se presta a ello.
Pero como también es imprescindible que un programador tenga la mente abierta y perceptiva a lo nuevo, creemos que es posible lograr ese encuentro de tendencias.
Un argumento muy utilizado en contra de la programación estructurada, es que las sentencias REM ocupaban mucho espacio y retrasaba la ejecución del programa.
Nada más antiguo que este argumento. En épocas pasadas cuando los ordenadores tenían diez veces menos capacidad de memoria, y eran cinco veces más lentos, había que eliminar todo lo superfluo, y lo que menos importancia tenía, era la sentencia REM. Por eso se formó el hábito de no escribirlas.

GENERALICEMOS EL BASIC

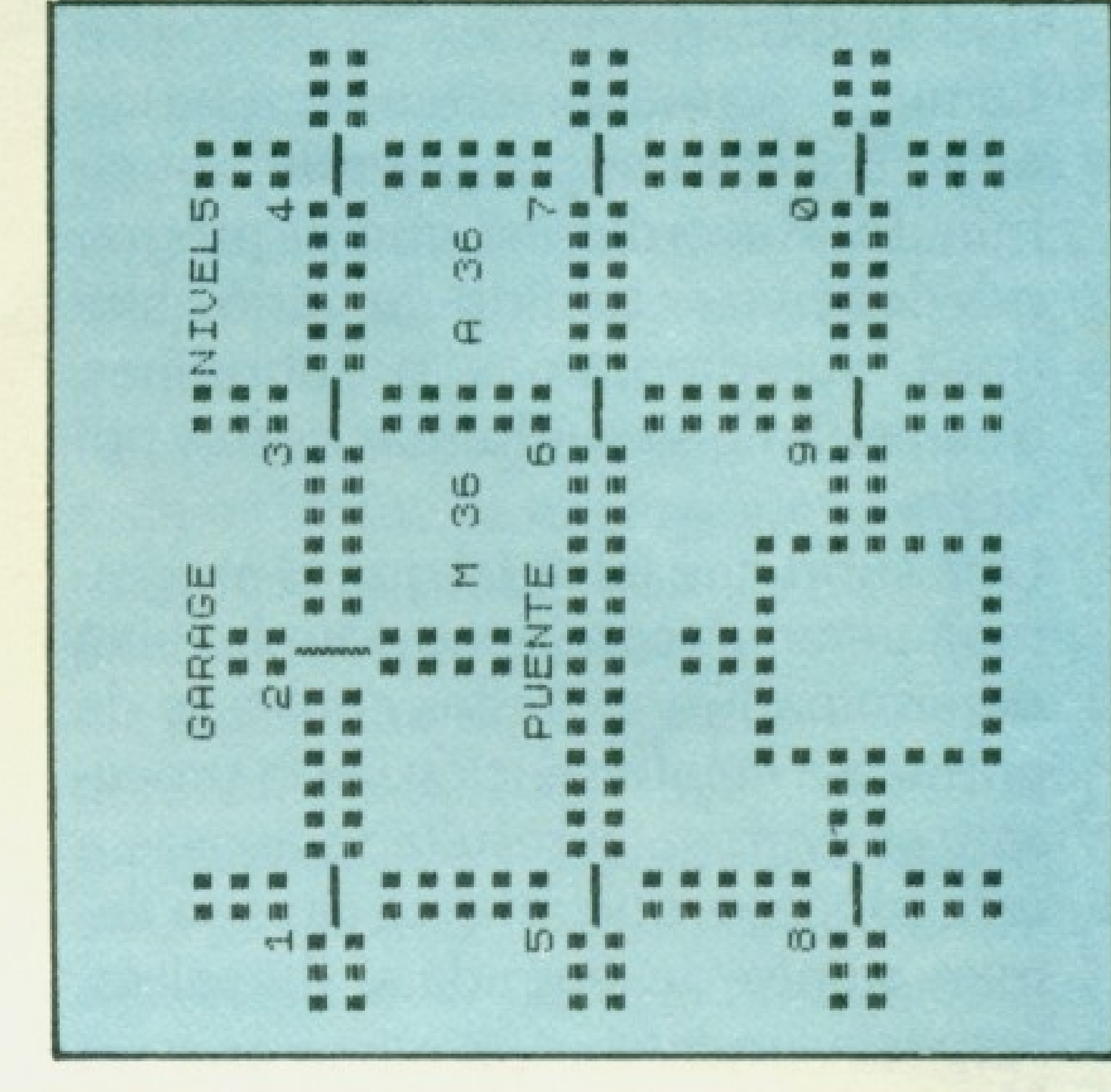
Existe un acuerdo entre los programadores en general, de numerar las líneas de programa de diez en diez. Esto es así porque una numeración de línea continua (1,2,3,...) no permitiría la inserción de nuevas líneas. Y esta necesidad se presenta en momentos inesperados.
También es conveniente, para hacer más claro y legible el programa, dejar líneas en blanco.

```
10 INPUT COSTO
20 :
30 PRINT COSTO * 1.4
```

Y como contribución a esta claridad, es bastante bueno acostumbrarse, a no superar un renglón cuando escribamos una línea de programa. Dejando, dentro de lo posible, lugar para utilizar un comentario aclaratorio, por medio del REMark.

PROGRAMAS CONTROL DE TRANSITO

COMP.: TS 2068/SPECTRUM/TK90X
CLAS.: JUEGO
AUTOR: Carlos Alberto Feito



El presente programa es un juego basado en la simulación de una torre de tránsito que controla la circulación de los automóviles de una ciudad.
Se incluyen en la misma un puente, un túnel y un garage que hacen más variado el juego.
El objeto es conseguir un flujo vehicular sin accidentes; que se producen cuando un automóvil intenta virar en una esquina hacia otra calle con el tránsito atascado hasta la bocacalle.
El número de automóviles crece con el tiempo, haciendo más difícil el control de los semáforos.
El resto de las explicaciones se las dará su ordenador personal.
El programa consta de dos partes, una basic y otra assembler.
Se debe cargar esta última, a partir de la posición de memoria 60000 y tiene una longitud de 2048 bytes.
Una vez copiadas ambas partes y antes de ejecutarlos, entren el comando GOTO 702 para tener una o varias copias de ellas en una cassette.

Estructura del programa

4-120: Presentación, comienzo y explicación.
125-220: Opción de nivel, inicialización de direcciones y cálculos asociados.
222-300: Dibuja la pantalla.
302-452: Sonido, lógica principal y fin.
600-612: Rutina de automóviles.

```
   1 REM
        **CONTROL DE TRANSITO**
         POR  C.A.F & J.P.B.

   2 LOAD ""CODE
   4 PAPER 0: BORDER 0: INK 7: C
LS
   6 PAUSE 50: FOR n=1 TO 5: PRI
NT AT n,5; PAPER 2; BRIGHT 1;"
   ": NEXT n
   8 PAUSE 150: FOR n=8 TO 12: P
RINT AT n,5; PAPER 6; BRIGHT 1;"
     ": NEXT n
  10 PAUSE 150: CLS
  12 PAUSE 5: FOR n=15 TO 19: PR
INT AT n,5; PAPER 4; BRIGHT 1;"
    ": NEXT n
  14 PAUSE 20: BEEP .3,-10: PAUS
E 10: BEEP .4,-10
  15 PRINT AT 17,12;"CONTROL DE
TRANSITO"
  16 PRINT AT 17,12; OVER 1;"^^^
""^^^ ^^ ""^^^^""""^"
  20 PAUSE 60: PRINT #0;"      P
RESIONE UNA TECLA"
  22 IF INKEY$<>"" THEN GO TO 22
  24 IF INKEY$="" THEN GO TO 24
 100 PAPER 5: BORDER 5: CLS
 102 PRINT AT 1,7; INK 6;"CONTRO
L DE TRANSITO"
 103 PRINT AT 1,7; OVER 1;"^^^""
^^^ ^^ ""^^^^""""^"
 104 INK 0: PRINT AT 5,1;"  UD.
ES EL CONTROLADOR DE LOS SEMAFOR
OS DE LA CIUDAD.         MEDIANT
E LAS TECLAS 1-O CONTROLAEL SEMA
FORO CORRESPONDIENTE.    PERO ..
.ATENCION!! ASI COMO LOS COCHES
ENTRAN AL GARAGE TAMBIEN SALEN D
E EL.                    SU MISI
ON ES LA DE REGULAR EL   TRANSIT
O DE MODO QUE NO OCURRA  UN ACCI
DENTE AL TRATAR DE VIRAR HACIA L
A IZQUIERDA EN UNA CALLE ATASCAD
A."
 106 PRINT #0;"     PULSE CUALQU
IER TECLA       "
 108 PAUSE 4E4
 110 PRINT AT 5,1;"  UD. DISPONE
 DE 5 NIVELES DE  DIFICULTAD CRE
CIENTES 1-5.        EL NUMERO MOME
NTANEO DE AUTOS ENTRANSITO SE IN
DICA CON A Y EL    NUMERO MAXIMO
CON M,EL QUE INDI CA EL PUNTAJE
FINAL."
 120 PRINT AT 10,20;"



                  "
 125 FOR N=0 TO 31: POKE (23264+
N),45: NEXT N
 130 PRINT #0;"         PULSE EL N
IVEL 1-5         "
 190 IF INKEY$="" THEN GO TO 190
 192 LET L$=INKEY$: IF L$<>"1" A
ND L$<>"2" AND L$<>"3" AND L$<>"
4" AND L$<>"5" THEN GO TO 188
 200 PAPER 4: BORDER 7: CLS : LE
T A=0: LET M=0: LET L=VAL L$: LE
T A$="  0": LET M$="  0"
 202 IF L=1 THEN LET P=25
 204 IF L=2 THEN LET P=20
 206 IF L=3 THEN LET P=15
 208 IF L=4 THEN LET P=10
 210 IF L=5 THEN LET P=5
 212 POKE 60000,0: POKE 60001,0:
 POKE 60002,0: POKE 60003,0: POK
E 60004,0: POKE 60005,1
 214 FOR n=60658 TO 60694 STEP 4
 216 LET x=(PEEK n)/2
 218 IF x=INT x THEN POKE n,2*x+
1
 220 NEXT n
 222 RANDOMIZE USR 60025
 224 PRINT AT 0,9; PAPER 6;"GARA
GE": PRINT AT 0,21; PAPER 6;"NIV
EL"
 226 PRINT AT 0,26; PAPER 7; BRI
GHT 1;L
 228 PRINT AT 2,2; PAPER 6; BRIG
HT 1;"1";AT 2,10;"2";AT 2,18;"3"
;AT 2,26;"4"
 230 PRINT AT 7,14; PAPER 6; BRI
GHT 1;"M";: PRINT TAB 15; PAPER
5; BRIGHT 1;M$: PRINT AT 7,22; P
APER 6; BRIGHT 1;"A";: PRINT TAB
 23; PAPER 5; BRIGHT 1;A$
 232 PRINT AT 9,2; PAPER 6; BRIG
HT 1;"5";AT 9,9; BRIGHT 0;"PUENT
E";AT 9,18; BRIGHT 1;"6";AT 9,26
;"7"
 236 PRINT AT 16,2; PAPER 6; BRI
GHT 1;"8";AT 16,18;"9";AT 16,26;
"0"
 300 IF INKEY$<>"" THEN LET A=1
 302 LET x=USR 60189
 304 IF PEEK 60004=0 THEN GO TO
350
 306 POKE x,248: BEEP .75,7: BEE
P .25,8: BEEP .5,7: BEEP .5,5: B
EEP .5,3: BEEP .5,2: BEEP 1,0
 308 PRINT AT 7,14; PAPER 6; BRI
GHT 1; FLASH 1;"M"
 310 PRINT #0;"    PARA NUEVO JU
EGO ENTER"
 312 IF INKEY$<>"" THEN GO TO 31
2
 314 IF INKEY$="" THEN GO TO 314
 316 LET L$=INKEY$: IF L$<>CHR$
13 THEN GO TO 312
 318 GO TO 100
 350 IF A=1 THEN BEEP .05,M/5: L
ET A=0
 352 FOR n=1 TO p: NEXT n
 354 RANDOMIZE USR 60464
 356 GO SUB 600
 402 RANDOMIZE USR 60538
 404 GO SUB 600
 450 FOR N=1 TO P: NEXT N
 452 GO TO 300
 600 LET a=PEEK 60000+256*PEEK 6
0001
 602 LET m=PEEK 60002+256*PEEK 6
0003
 604 LET a$=STR$ a: LET m$=STR$
m
 606 IF LEN a$<3 THEN LET a$=" "
+a$: GO TO 606
 608 IF LEN m$<3 THEN LET m$=" "
+m$: GO TO 608
 610 PRINT AT 7,15;m$;AT 7,23;a$
 612 RETURN
 702 SAVE "CONTROL DE" LINE 1: S
AVE "TRAFICO"CODE 60000,2048: GO
 TO 702
```

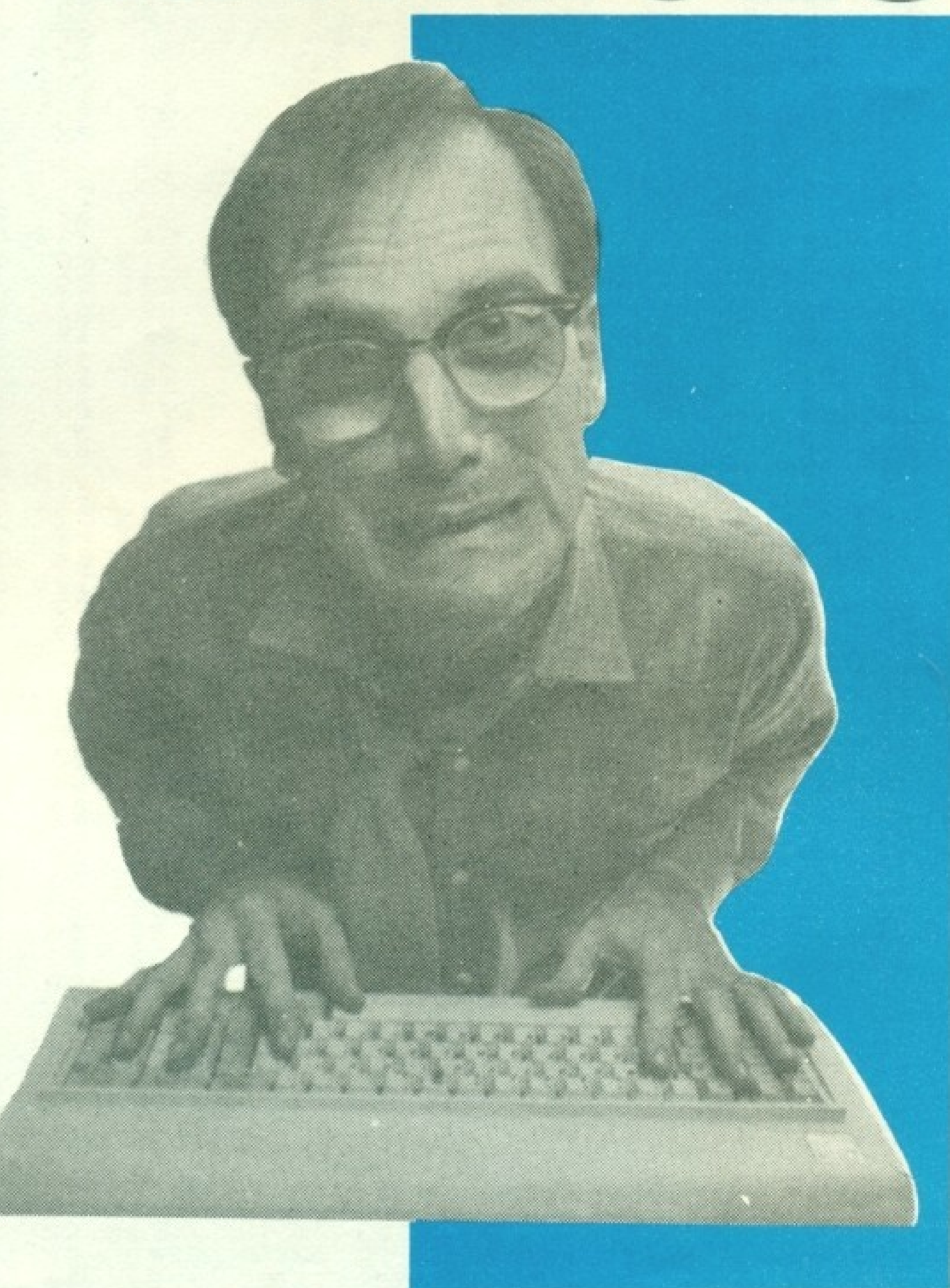
```
2    0    2    0    0    1    32   1
35   67   98   64   2    35   98   97
64   1    32   1    35   67   98   64
0    221  33   241  236  221  126  0
254  0    40   81   221  110  2    221
102  3    35   35   221  126  1    230
192  229  205  87   242  225  1    30
0    9    229  205  87   242  225  1
35   0    9    229  205  87   242  225
1    30   0    9    229  205  87   242
225  1    97   0    167  237  66   229
1    33   0    9    54   40   35   54
40   1    31   0    9    54   40   35
54   40   225  205  231  241  1    4
0    221  9    24   168  221  110  1
221  102  2    221  126  3    205  87
242  1    4    0    221  9    221  126
1    254  0    40   19   221  110  0
221  102  1    221  126  4    205  87
242  1    5    0    221  9    24   230
221  35   221  35   221  126  1    254
0    200  221  110  0    221  102  1
221  126  6    205  87   242  1    7
0    221  9    24   231  221  33   241
236  221  126  0    254  0    40   56
58   4    92   221  190  0    40   7
1    4    0    221  9    24   234  221
110  2    221  102  3    221  126  1
230  1    254  0    40   14   221  126
1    230  192  205  36   242  221  203
1    134  24   12   221  126  1    230
192  205  231  241  221  203  1    198
221  33   241  236  221  126  0    254
0    200  0    0    221  110  2    221
102  3    17   6    0    221  126  1
230  1    254  0    40   3    17   0
0    221  229  221  33   102  234  221
25   221  126  0    50   114  234  221
126  3    50   117  234  58   120  92
203  71   40   14   221  126  1    50
115  234  221  126  2    50   116  234
24   12   221  125  1    50   116  234
221  126  2    50   115  234  58   120
92   203  71   40   14   221  126  4
50   118  234  221  126  5    50   119
234  24   12   221  126  4    50   119
234  221  126  5    50   118  234  221
33   114  234  221  126  0    254  0
40   60   221  94   0    22   0    229
25   68   77   126  254  40   32   38
225  229  221  94   1    22   0    25
126  254  40   40   7    54   40   62
45   2    24   18   225  229  221  94
2    22   0    25   126  254  40   40
23   54   40   62   45   2    225  1
3    0    221  9    24   189  221  225
1    4    0    221  9    195  100  235
225  225  33   100  234  54   1    201
221  33   25   237  221  110  1    221
102  2    126  254  40   32   9    54
45   42   96   234  43   34   96   234
1    4    0    221  9    221  126  1
254  0    200  0    221  110  0    221
102  1    126  254  40   32   13   229
221  110  2    221  102  3    126  254
40   32   8    225  1    5    0    221
9    24   218  54   40   225  54   45
24   242  221  33   157  241  221  126
1    254  0    40   66   221  110  0
221  102  1    221  78   2    221  70
3    221  94   4    221  86   5    126
254  40   32   36   58   120  92   203
71   40   5    10   254  40   32   18
26   254  40   40   8    54   45   213
225  54   40   24   11   10   254  40
40   6    54   45   197  225  54   40
1    7    0    221  9    24   183  221
110  2    221  102  3    126  254  40
40   25   54   40   42   96   234  35
34   96   234  237  75   98   234  167
237  66   250  235  236  42   96   234
34   98   234  33   100  234  54   0
201  48   129  26   90   49   1    66
88   50   1    74   88   51   1    82
88   52   1    90   88   53   65   34
89   54   65   50   89   55   65   58
```

```
89   56   129  2    90   57   129  18
90   0    43   88   0    44   88   76
88   0    75   88   43   88   0    203
88   171  88   0    49   90   50   90
128  70   90   69   90   128  234  89
235  89   64   15   90   47   90   128
104  90   72   90   128  105  88   106
88   0    142  88   141  88   0    134
88   133  88   0    113  88   114  88
0    36   88   68   88   0    52   88
84   88   0    60   88   92   88   0
97   88   98   88   0    195  88   163
88   0    150  88   149  88   0    121
88   122  88   0    158  88   157  88
0    219  88   187  88   0    211  88
179  88   0    4    89   36   89   64
20   89   52   89   64   28   89   60
89   64   65   89   66   89   64   81
89   82   89   64   89   89   90   89
64   102  89   101  89   64   118  89
117  89   64   126  89   125  89   64
163  89   131  89   64   179  89   147
89   64   187  89   155  89   64   228
89   4    90   64   244  89   20   90
64   252  89   28   90   64   33   90
34   90   128  86   90   85   90   128
57   90   58   90   128  94   90   93
90   128  131  90   99   90   128  147
90   115  90   128  155  90   123  90
128  235  88   203  88   0    48   90
49   90   128  71   90   70   90   128
233  89   234  89   64   239  89   15
90   64   136  90   104  90   128  104
88   105  88   0    143  88   142  88
0    135  88   134  88   0    112  88
113  88   0    4    88   36   88   0
20   88   52   88   0    28   88   60
88   0    96   88   97   88   0    227
88   195  88   0    151  88   150  88
0    120  88   121  88   0    159  88
158  88   0    251  88   219  88   0
243  88   211  88   0    228  88   4
89   0    244  88   20   89   0    252
88   28   89   0    64   89   65   89
64   80   89   81   89   64   79   89
80   89   64   88   89   89   89   64
103  89   102  89   64   104  89   103
89   64   119  89   118  89   64   127
89   126  89   64   195  89   163  89
64   211  89   179  89   64   219  89
187  89   64   196  89   228  89   64
212  89   244  89   64   220  89   252
89   64   32   90   33   90   128  238
89   239  89   64   237  89   238  89
64   87   90   86   90   128  56   90
57   90   128  78   89   79   89   64
105  89   104  89   64   95   90   94
90   128  168  90   136  90   128  169
90   168  90   128  163  90   131  90
128  236  89   237  89   64   179  90
147  90   128  187  90   155  90   128
170  90   169  90   128  164  90   4
88   128  180  90   20   88   128  11
89   235  88   64   188  90   28   88
128  3    88   163  90   0    232  89
233  89   64   19   88   179  90   0
171  90   170  90   128  27   88   187
90   0    128  88   159  88   0    127
88   96   88   0    77   89   78   89
64   106  89   105  89   64   204  89
236  89   64   95   89   64   89   64
96   89   127  89   64   76   89   77
89   64   107  89   106  89   64   171
89   11   89   64   8    90   232  89
128  64   90   95   90   128  63   90
32   90   128  172  90   171  90   128
172  89   204  89   64   40   90   8
90   128  35   88   3    88   0    51
88   19   88   0    59   88   27   88
0    75   89   76   89   64   108  89
107  89   64   129  88   128  88   0
103  88   104  88   0    136  88   135
88   0    111  88   112  88   0    144
88   143  88   0    119  88   120  88
0    152  88   151  88   0    126  88
127  88   0    203  89   171  89   64
```

```
173  90   172  90   128  55   90   56
90   128  88   90   87   90   128  132
90   164  90   128  12   89   172  89
64   102  88   103  88   0    110  88
111  88   0    118  88   119  88   0
137  88   136  88   0    145  88   144
88   0    153  88   152  88   0    74
89   75   89   64   109  89   108  89
64   174  90   173  90   128  148  90
180  90   128  156  90   168  90   128
39   90   40   90   128  87   89   88
89   64   120  89   119  89   64   27
89   251  88   64   220  88   252  88
0    94   89   95   89   64   3    89
227  88   64   196  88   228  88   0
19   89   243  88   64   212  88   244
88   0    175  90   174  90   128  227
89   195  89   64   164  89   196  89
64   243  89   211  89   64   143  90
175  90   128  180  89   212  89   64
251  89   219  89   64   188  89   220
89   64   67   88   35   88   0    236
88   12   89   0    111  90   143  90
128  83   88   51   88   0    91   88
59   88   0    38   90   39   90   128
97   89   96   89   64   130  88   129
35   0    125  88   126  88   0    86
69   87   89   64   73   89   74   89
64   110  89   109  89   64   121  89
120  89   64   65   90   64   90   128
101  88   102  88   0    72   89   73
89   64   111  89   110  89   64   164
88   196  88   0    109  88   110  88
0    79   90   111  90   128  71   89
72   89   64   112  89   111  89   64
138  88   137  88   0    204  88   236
88   0    146  88   145  88   0    117
88   118  88   0    180  88   212  88
0    80   90   79   90   128  154  88
153  88   0    54   90   55   90   128
89   90   88   90   128  70   89   71
89   64   113  89   112  89   64   188
88   220  88   0    172  88   204  88
0    35   89   3    89   64   132  89
164  89   64   98   89   97   89   64
81   90   80   90   128  51   89   19
89   64   148  89   180  89   64   85
89   86   89   64   114  89   113  89
64   59   89   27   89   64   93   89
94   89   64   156  89   188  89   64
122  89   121  89   64   62   90   63
90   128  69   89   70   89   64   3
90   227  89   128  100  90   132  90
128  66   90   65   90   128  37   90
38   90   128  116  90   148  90   128
19   90   243  89   128  82   90   81
90   128  53   90   54   90   128  124
90   156  90   128  27   90   251  89
128  90   90   89   90   128  61   90
62   90   128  0    0    72   90   71
90   40   90   128  47   90   48   90
79   90   128  235  89   203  89   236
89   64   0    0    44   88   254  128
40   14   254  64   40   5    17   0
24   24   8    17   0    17   24   3
17   0    10   167  237  82   201  54
0    36   54   0    36   54   60   36
54   60   36   54   60   36   54   60
36   54   0    36   54   0    201  1
33   0    9    205  182  241  229  62
7    54   0    36   61   32   250  54
255  225  35   229  62   7    54   0
36   61   32   250  54   255  225  11
11   9    229  54   255  36   62   7
54   0    36   61   32   250  225  35
54   255  36   62   7    54   0    36
61   32   250  201  1    33   0    9
205  182  241  229  62   8    54   1
36   61   32   250  225  35   229  62
8    54   128  36   61   32   250  225
11   11   9    229  62   8    54   1
36   61   32   250  225  35   62   8
54   128  36   61   32   250  201  54
45   205  182  241  205  207  241  201
0
```

MATRICES

Este artículo va dirigido a estudiantes secundarios, de primer año de la universidad y a todos aquellos que aunque estén más avanzados, necesiten programar la carga y operación de distintos tipos de matrices e interpretar el modo de hacerlo.



Con el advenimiento de las micropersonales el aprendizaje de las matemáticas se ve facilitado, ya que con un poco de dominio del BASIC y otro poco de curiosidad, podemos interpretarlas más rápidamente, sin perder tiempo en engorrosos cálculos, los que tendremos bien conceptuados al enseñarle a la computadora cómo debe hacerlos (qué mejor método para aprender que enseñando).

Una vez comprendido determinado tema, el profesor debe ser un órgano de consulta, el afán investigativo del alumno debe hacer el resto.

Las micro desarrollan plenamente esta inquietud por investigar, ya que en un programa desarrollado por nosotros es un desafío tomar el camino lógico correcto, corrigiéndonos a nosotros mismos.

Este artículo va dirigido a estudiantes de los últimos años del ciclo secundario, primer año de la universidad, etc. y a todos aquellos que aunque estén más avanzados, necesiten programar la carga y operación de distintos tipos de matrices e interpretar el modo de hacerlo.

Cualquier curso de matemáticas moderno, debe tratar el álgebra de matrices como una de sus secciones más importantes.

Las aplicaciones en los distintos campos son enormes: Física, Química, Ingeniería, por ejemplo los niveles de energía de un electrón y muchas constantes físicas, pasan a ser calculadas a partir de ciertas matrices de infinitas filas por infinitas columnas, así como el estudio de las perturbaciones del movimiento de la luna. Pueden hallarse matrices en el cálculo de estructuras y resistencia de materiales; cálculo de órbitas astronómicas; cálculos estadísticos, tratamiento de cadenas de producción, problemas de psicología y sociología: por ejemplo matrices de comunicación que nos dan conclusiones sobre el dominio de ciertos grupos sobre otros, regla de formación de matrimonios, o sea que por el cálculo matricial puede resolverse la intrincada maraña de las relaciones de parentesco de las sociedades primitivas (Antropología). Comportamiento del Mercado, teoría de los juegos de estrategia que a su vez es una rama de la investigación operacional para la toma de decisiones, por ejemplo en acciones de guerra (o civiles). Genética, Teoría Cuántica, Teoría de las vibraciones, Técnicas del Computador Moderno...; es decir, en la aplicación de ideas contemporáneas para la solución de problemas contemporáneos.

A este tema lo desarrollaremos en dos partes: en este número trataremos la carga de matrices, suma y resta, multiplicación de una matriz por un escalar, producto de matrices, estableciendo sus propiedades; en el próximo número, programas para resolución de determinantes, Matriz traspuesta, Matriz adjunta, Matriz inversa. Cabe recordar que estos programas son utilizables por aquellas micro compatibles con el SPECTRUM.

Lamentablemente esta micro-personal no carga una Matriz directamente como lo haría una computadora profesional y menos aún realiza operaciones (directamente), por eso debemos proceder a cargarla nosotros por medio de un programa; lo primero que hacemos es dimensionarla: recordemos que una matriz es una disposición (arreglo) de números por lo que su nombre debe ser una letra sin el signo $ (con signo $ sería una matriz de cadena que no es el tema que estamos tratando) y cuyos elementos están dispuestos en forma rectangular definidos por su ubicación (fila I, columna J), tendrá entonces una dimensión de tantas filas por tantas columnas, en primer término siempre pondremos la fila, en segundo siempre la columna, ejemplo DIM A(3,4) tendrá 3 filas por 4 columnas.

Un Vector es un caso particular de una matriz de una fila o de una columna.

La carga la podemos realizar con INPUT, READ, LET. Como ejemplo cargaremos una matriz de 3 filas por 4 columnas.

```
10 DIM A(3,4)
20 FOR I =1 TO 3
```

En la primer vuelta de este lazo FOR-NEXT estamos en la fila 1, imprimimos de qué fila se trata, y con el próximo lazo por cada fila cargamos las columnas.

```
30 PRINT "FILA =";I
40 FOR J =1 TO 4
50 READ A(I,J)
```

Si ponemos 55 PAUSE 0 cada vez que pulsemos ENTER, veríamos ingresar uno por uno a los datos (elementos) de la Matriz.

```
60 PRINT A(I,J)
70 NEXT J
```

Cuando I =1 aquí ya cargó una fila

completa, luego va a 20 y comienza con la fila siguiente.

```
80 PRINT
90 NEXT I
100 DATA 2,1, -1,0,4,3,2,5,3, -2,0,7
```

La cantidad de datos debe coincidir exactamente con la cantidad de elementos de la matriz dimensionada (en este caso 3x4 =12 elementos).

Existe la tentación de ver la Matriz representada en pantalla, distribuida por filas y columnas, ello se podría lograr si cada elemento de la matriz no tiene más de 3 ó 4 dígitos y la cantidad de columnas no supera 6, la cantidad de filas no interesarían ya que harían un SCROLL, si este es el caso haríamos en 60 PRINT A(I,J); " "; (se corregiría el espacio en función de la cantidad de columnas).

Las operaciones a realizar podrían ser para comenzar sumas y restas. Para sumar 2 matrices nos faltaría cargar otra matriz B, para ello utilizaremos el mismo procedimiento anterior. La suma de una matriz A con otra B nos dará una nueva matriz C y como sumar 2 matrices es sumar cada uno de los elementos correspondientes a la misma fila y columna, tendrá la misma dimensión, le indicaremos a nuestra micro cómo debe hacer la operación: tengamos en cuenta que no se permite la suma o resta de más de 2 matrices por vez y que ambas deben tener la misma dimensión.

```
300 DIM C(3,4)
310 FOR I =1 TO 3
320 PRINT "FILA =";I
330 LET C(I,J) = A(I,J) + B(I,J)
```

Estamos utilizando LET para cargar la matriz

```
340 PRINT C(I,J)
```

Imprimimos, si no aparecen los resultados por pantalla.

```
350 NEXT J
360 PRINT
```

Este PRINT es sólo para separar

```
370 NEXT I
```

Recordemos las propiedades de la suma de matrices

(A+B)+C = A+(C+B) Asociativa
A+B = B+A Conmutativa
A+0 =0+A =A Existencia de la matriz nula (Todos sus elementos =0)
A+(−A) = (−A)+A = 0

El criterio para la resta es exactamente el mismo.

Multiplicación escalar

k =constante

En este caso todos los elementos de una matriz se multiplican por una constante. Si tenemos una matriz cargada A y la queremos multiplicar por 3,2, obtendremos una nueva matriz de igual cantidad de filas por igual cantidad de columnas que la A, sólo que cada uno de sus elementos resulta de multiplicar a cada elemento de A por 3,2. Si no utilizáramos A en un futuro, nada nos impediría volcar esos nuevos elementos en la misma A, pero si no dimensionamos una nueva matriz D y la cargamos como hicimos con C (si usamos A, también) haciendo LET D(I,J) = 3,2 * A(I,J).

Tengamos en cuenta que esta constante puede ser una letra, que a la hora de ser requerida represente un número.

No se permite la multiplicación escalar por más de una matriz, la expresión LET H(I,J) =9 * A * B ó 18 * A + B es incorrecta.

Propiedades
k * (A+B) =k * A * k * B Distributiva
k =1 k*A = A

El panorama se complica un poco, cuando debemos enseñarle a nuestra micro, a multiplicar matrices. Tengamos en cuenta que:

1) Un producto matricial será posible si y sólo si siendo la matriz A el primer factor y B el segundo, el número de columnas de A sea el mismo que el número de filas de B.
2) No se permite la multiplicación de más de 2 matrices.
3) El producto de matrices no obedece la ley conmutativa, los productos AxB y BxA de matrices, no tienen que ser iguales.

Propiedades
(A*B)*C = A*(BxC) Asociativa
A*(B+C) = A*B + A*C Distributiva
(B+C)*A = B*A + C*A
k * (A*B) =(k*A) * B =A * (kxB) donde k es un escalar

Repasemos cómo se logra, manualmente, resolver el problema: dimensionaremos particularmente una matriz A(3,3) y otra B(3,2), cuya expresión general sería A(L,M) y B(M,N), el producto de ambas dará como resultado una nueva matriz C(3,2) admitiendo la forma general de C(L,N).

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

$$B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix}$$

AxB obtenemos $C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \\ c_{31} & c_{32} \end{pmatrix}$

Cada elemento de C se logra haciendo:

$c_{11} = a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31}$
$c_{12} = a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32}$
$c_{21} = a_{21} * b_{11} + a_{22} * b_{22} + a_{23} * b_{31}$
$c_{22} = a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32}$
$c_{31} = a_{31} * b_{11} + a_{32} * b_{21} + a_{33} * b_{31}$
$c_{32} = a_{31} * b_{12} + a_{32} * b_{22} + a_{33} * b_{32}$

Si encolumnamos la operación para la obtención de uno cualquiera de los elementos de C por ejemplo c_{12}, podremos analizar sus posiciones:

a(1,1) * b (1,2)
+a(1,2) * b (2,2)
a(1,3) * b (3,2)

Observamos que dos números, si bien distintos, se mantienen constantes (fila 1 y columna 2), a ese valor que cambia cada vez (lo llamaremos G), nos obliga a un nuevo FOR-NEXT, entonces el patrón de variación (expresión general), será:

a(I,G) * b(G,J)

Suponiendo tener cargadas las matrices A y B (que reúnen la condición 1), ,), un programa general que cargue una matriz C (resultado de AxB) sería

```
500 DIM C(L,N)
```

Recordemos A(L,M) y B(M,N) por eso ese dimensionamiento de C (debe quedar claro que un programa será usado en forma general, los dimensionamientos particulares que hemos hecho sólo servían de comparación para aclarar conceptos).

```
510 FOR I =1 TO L
520 PRINT " FILA =";I
530 FOR J = 1 TO N
540 LET S1 = 0 ; LET S = 0
```

Como hay sumas y multiplicaciones interrelacionadas necesitamos estos 2 acumuladores

```
550 FOR G = 1 TO M
```

En un producto M =cantidad de columnas del primer factor =cantidad de filas del segundo factor.

```
560 LET S1 = A(I,G) x B(G,J)
570 LET S = S+S1
580 NEXT G
590 LET C(I,J) = S
600 PRINT C(I,J)
610 NEXT J
620 NEXT I
```

Héctor Raúl Mansilla

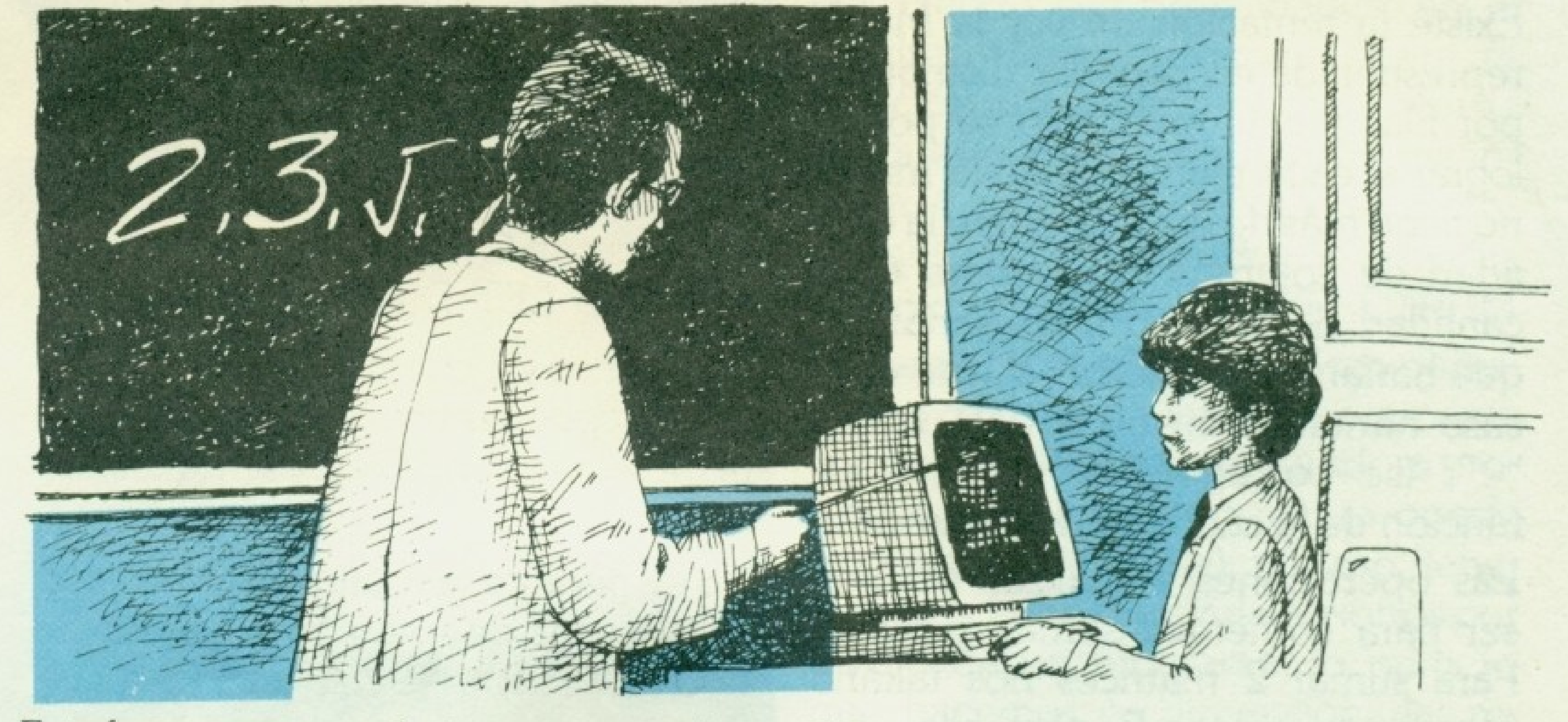
PROGRAMAS REDUCCION DE MATRICES

COMP.: CZ 1000/1500; TK 83/85
CONF.: 16 K
CLAS.: UTILITARIO
AUTOR: Pablo Ignacio Bazán

Comentarios previos

Este utilitario tiene por finalidad reducir por filas, matrices de "m" filas y "n" columnas.
Es un soft muy útil para los estudiantes de carreras técnicas, de nivel terciario y secundario.
Luego de una breve presentación, el programa requiere el ingreso de los datos correspondientes a la matriz (cantidad de filas y columnas), no pudiendo exceder los 100 elementos.
Para matrices M que pertenecen a un cuerpo K de m filas por n columnas, con $m > 8$ tal que n 6, veremos la matriz en pantalla de la forma M(I,J) = X, con $m < 9$ y $n < 7$ se verá en pantalla con la forma de matriz.
Los coeficientes (los elementos de la matriz), deben estar comprendidos entre -999 y 9999, esto es:
$-1000 < X < 1000$
Una vez ingresados los elementos, el programa comienza a trabajar de la siguiente forma:
* Ordena la matriz (líneas 2000-3000)
* Hace cero la columna del primer elemento conductor (50-140)
* Ordena esta otra matriz (2000-3000)
* Hace cero la columna del segundo elemento conductor (50-140)
* Ordena
* ...
* y así hasta el último elemento conductor.

Esta tarea se hace muy rápido (sobre todo si la matriz no tiene muchas filas), por lo general tarda 15" aprox.
Una vez hecha la reducción de la matriz, comienza con la visualización de ésta.
Al imprimir en la pantalla la matriz, se va fijando si los elementos de ésta son enteros, si es así los imprime tal cual. Si no lo son se ejecuta una subrutina que imprime el número en forma de fracción, simplificando a ambos números si es posible por 2, 3, 5, 7, 11, 13, 17 ó 19.



Esta forma es muy práctica aunque tarda unos 6" en ponerlos como fracción.
La visualización en 'forma de matriz' se complica cuando son números muy grandes pudiendo quedar encimados, por eso al terminar de imprimir tenemos la opción de verla de la forma M(I,J) = X, pulsando la letra 'M'. Para terminar con esta, pulsamos cualquier tecla y el programa pedirá una nueva matriz.

Utilidad

Uno de sus mayores usos es el de resolver sistemas de m ecuaciones con n incógnitas.
Por ejemplo, consideremos el siguiente sistema de ecuaciones lineales:

$$
\begin{array}{l}
5X1 + 0X2 + 15X3 + 12X4 + 6X5 + 0X6 = 0 \\
3X1 + 2X2 + 0X3 + 3X4 + 1X5 + 0X6 = 5 \\
5X1 + 1X2 + 0X3 + 5X4 + 2X5 + 9X6 = 3
\end{array}
$$

y queremos encontrar una solución general y una particular.
De acuerdo a este sistema tendremos 2 matrices:
1) La matriz de coeficientes

$$
A = \begin{array}{cccccc} 5 & 0 & 15 & 12 & 6 & 0 \\ 3 & 2 & 0 & 3 & 1 & 0 \\ 5 & 1 & 0 & 5 & 2 & 9 \end{array}
$$

2) La matriz de términos independientes

$$
H = \begin{array}{c} 0 \\ 5 \\ 3 \end{array}
$$

La matriz que debemos tomar en cuenta entonces, es la matriz ampliada: M(A/H)

$$
M = \begin{array}{ccccccc} 5 & 0 & 15 & 12 & 6 & 0 & 0 \\ 3 & 2 & 0 & 3 & 1 & 0 & 5 \\ 5 & 1 & 0 & 5 & 2 & 9 & 3 \end{array}
$$

Entonces cuando la máquina nos pregunte cuántas filas tiene la matriz, responderemos 3, y cuando lo haga por las columnas, responderemos 7; luego entraremos la matriz M coeficiente a coeficiente.
Después de unos segundos, la máquina nos dará la matriz reducida:

$$
\begin{array}{ccccccc} 1 & 0 & 0 & 1 & 3/7 & 18/7 & 1/7 \\ 0 & 1 & 0 & 0 & 1/7 & -27/7 & 16/7 \\ 0 & 0 & 1 & 7/15 & 9/35 & -6/7 & -1/21 \end{array}
$$

Esto significa el siguiente sistema

$$
\begin{array}{l}
1X1 + 0X2 + 0X3 + 1X4 + 3/7\,X5 + 18/7\,X6 = 1/7 \\
0X1 + X2 + 0X3 + 0X4 + 1/7\,X5 - 27/7\,X6 = 16/7 \\
0X1 + 0X2 + X3 + 7/15\,X4 + 9/35\,X5 - 6/7\,X6 = 1/21
\end{array}
$$

De donde podemos despejar X1, X2 y X3, facilitando de este modo la obtención de las soluciones.

```
  1 GOSUB 9500
 10 GOSUB 7000
 30 FAST
 40 GOSUB 2000
 50 FOR G=1 TO F
 60 FOR H=1 TO F
 70 IF H=G THEN GOTO 120
 75 IF K(G)+1>C THEN LET H=F
 76 IF K(G)+1>C THEN GOTO 120
 80 IF B(H,K(G)+1)=0 THEN GOTO 120
 90 FOR D=1 TO C
100 LET C(H,D)=B(G,K(G)+1)*B(H,D)-B(H,
    K(G)+1)*B(G,D)
110 NEXT D
111 FOR D=1 TO C
112 LET B(H,D)=C(H,D)
113 NEXT D
116 GOSUB 2000
```

```
117 LET H=H-1
120 NEXT H
140 NEXT G
142 GOSUB 2000
145 SLOW
200 IF C<7 AND F<9 THEN LET PIL=0
220 IF C>=7 OR F>=9 THEN LET PIL=1
230 LET TAB=0
300 IF PIL=1 THEN GOTO 600
350 PRINT AT 11,0;"MATRIZ REDUCIDA:"
400 FOR G=1 TO F
405 IF K(G)+1>C THEN GOTO 434
410 FOR D=1 TO C
420 LET C(G,D)=B(G,D)/B(G,K(G)+1)
422 IF C(G,D)<>INT C(G,D) THEN GOTO 6000
425 PRINT AT 12+G,(D-1)*5;C(G,D)
430 NEXT D
432 GOTO 440
434 FOR U=1 TO C
435 PRINT AT 12+G,(U-1)*5;B(G,U)
436 NEXT U
440 NEXT G
450 PRINT AT 20,0;"""M"" PARA VERLA MAS CLARA"
500 PRINT AT 21,0;"apriete una tecla para otra"
510 IF INKEY$="" THEN GOTO 510
512 LET S$=INKEY$
515 CLS
516 IF S$="M" THEN GOTO 600
520 RUN 10
600 PRINT AT 0,0;"MATRIZ REDUCIDA:"
601 LET TAB=0
602 FOR G=1 TO F
605 IF K(G)+1>C THEN GOTO 690
610 FOR D=1 TO C
620 LET C(G,D)=B(G,D)/B(G,K(G)+1)
630 IF C(G,D)<>INT C(G,D) THEN GOTO 6000
640 PRINT AT D+1,TAB;"M(";G;",";D;")=";C(G,D)
650 NEXT D
660 LET TAB=TAB+16
670 IF TAB=32 THEN GOTO 750
680 GOTO 720
690 FOR U=1 TO C
700 PRINT AT U+1,TAB;"M(";G;",";U;")=";B(G,U)
710 NEXT U
715 GOTO 660
720 NEXT G
730 GOTO 450
750 PRINT AT 21,0;"apriete una tecla"
777 IF G=F THEN GOTO 450
800 IF INKEY$="" THEN GOTO 800
810 CLS
820 LET TAB=0
830 NEXT G
900 GOTO 450
2000 FOR J=1 TO F
2001 LET K(J)=0
2002 NEXT J
2009 FOR A=1 TO F
2010 FOR N=1 TO C
2020 IF B(A,N)=0 THEN LET K(A)=K(A)+1
2030 IF B(A,N)=0 THEN GOTO 2050
2040 LET N=C
2050 NEXT N
2060 NEXT A
2500 FOR I=1 TO F
2510 IF I+1>F THEN GOTO 2580
2520 FOR J=1+I TO F
2530 IF K(I)<=K(J) THEN GOTO 2570
2540 FOR L=1 TO C
2542 LET PIB=B(J,L)
2545 LET B(J,L)=B(I,L)
2547 LET B(I,L)=PIB
2550 NEXT L
2555 LET O=K(I)
2560 LET K(I)=K(J)
2565 LET K(J)=O
2570 NEXT J
2580 NEXT I
3000 RETURN
6000 FAST
6002 LET N1=B(G,D)
6005 LET N2=B(G,K(G)+1)
6006 FOR Q=1 TO 9
6007 LET X(Q)=0
6008 LET Y(Q)=0
6009 NEXT Q
6010 LET T$=""+CHR$ 11+"$)<"
6020 FOR Z=1 TO 8
6030 IF N1/CODE T$(Z)=INT (N1/CODE T$(Z)) THEN
     LET X(Z)=X(Z)+1
6040 IF N1/CODE T$(Z)<>INT (N1/CODE T$(Z))
     THEN GOTO 6070
6050 LET N1=N1/CODE T$(Z)
6060 GOTO 6030
6070 NEXT Z
6080 LET X(Z)=N1
6100 FOR Z=1 TO 8
6110 IF N2/CODE T$(Z)=INT (N2/CODE T$(Z)) THEN
     LET Y(Z)=Y(Z)+1
6120 IF N2/CODE T$(Z)<>INT (N2/CODE T$(Z))
     THEN GOTO 6150
6130 LET N2=N2/CODE T$(Z)
6140 GOTO 6110
6150 NEXT Z
6160 LET Y(Z)=N2
6200 FOR I=1 TO 8
6210 IF X(I)<Y(I) THEN GOTO 6250
6220 LET X(I)=X(I)-Y(I)
6230 LET Y(I)=0
6240 GOTO 6270
6250 LET Y(I)=Y(I)-X(I)
6260 LET X(I)=0
6270 NEXT I
```

```
6280 LET M=(2**X(1))*(3**X(2))*(5**X(3))*(7**X(4))*(11**X(5))*(13**X(6))*(17**X(
7))*(19**X(8))*X(9)
6290 LET N=(2**Y(1))*(3**Y(2))*(5**Y(3))*(7**Y(4))*(11**Y(5))*(13**Y(6))*(17**Y(
7))*(19**Y(8))*Y(9)
6300 SLOW
6325 IF S$="M" THEN GOTO 6460
6350 IF PIL=1 THEN GOTO 6460
6400 IF M<0 AND N<0 THEN PRINT AT 12+G,((D-1)*5)-1;-M;"/";-N
6425 IF M>0 AND N<0 THEN PRINT AT 12+G,((D-1)*5)-1;-M;"/";-N
6450 IF (M>0 AND N>0) OR (M<0 AND N>0) THEN PRINT AT 12+G,((D-1)*5)-1;M;"/";N

6455 GOTO 430
6460 IF M<0 AND N<0 THEN PRINT AT D+1,TAB;"M(";G;",";D;")=";-M;"/";-N

6470 IF M>0 AND N<0 THEN PRINT AT D+1,TAB;"M(";G;",";D;")=";-M;"/";-N

6480 IF (M>0 AND N>0) OR (M<0 AND N>0) THEN PRINT AT D+1,TAB;"M(";G;",";D;")=";M
;"/";N
6500 GOTO 650
7002 PRINT AT 0,0;
7005 PRINT "CANTIDAD DE FILAS ? ";
7006 INPUT F
7007 PRINT ;F,"CANTIDAD DE COLUMNAS ? ";
7008 INPUT C
7009 PRINT C
7010 IF F*C>100 THEN GOTO 8200
7020 PRINT ,,"ENTRE MATRIZ NUMERO POR NUMERO"
7022 PRINT ,,,,"ANOTE LA MATRIZ                          ANTES DE ENTRARLA"

7025 LET S$=""
7030 DIM X(10)
7040 DIM Y(10)
7050 LET FBT=0
7060 DIM C(F,C)
7070 DIM M$(F*C,4)
7080 DIM B(F,C)
7090 DIM K(F)
7092 PRINT AT 21,0;"apriete una tecla"
7094 IF INKEY$="" THEN GOTO 7094
7096 CLS
7098 PRINT AT 0,0;"MATRIZ:"
7100 IF C<7 AND F<9 THEN GOTO 8000
7110 LET TAB=0
7120 FOR N=1 TO F
7130 FOR V=1 TO C
7140 LET FBT=FBT+1
7150 INPUT M$(FBT)
7160 IF M$(FBT)="    " OR M$(FBT)="-   " THEN GOTO 7150
7170 PRINT AT V+1,TAB;"M(";N;",";V;")=";M$(FBT)
7180 LET B(N,V)=VAL M$(FBT)
7190 NEXT V
7200 LET TAB=TAB+16
7210 IF TAB=32 THEN GOTO 7235
7220 NEXT N
7230 GOTO 7280
7235 PRINT AT 21,0;"apriete una tecla"
7240 IF INKEY$="" THEN GOTO 7240
7250 CLS
7260 LET TAB=0
7270 NEXT N
7280 CLS
7300 RETURN
8000 PRINT AT 0,0;"MATRIZ:"
8010 FOR N=1 TO F
8020 FOR V=1 TO C
8030 LET FBT=FBT+1
8040 INPUT M$(FBT)
8050 IF M$(FBT)="    " OR M$(FBT)="-   " THEN GOTO 8040
8060 PRINT AT N+1,(V*5)-LEN STR$ VAL M$(FBT);M$(FBT)
8070 LET B(N,V)=VAL M$(FBT)
8080 NEXT V
8090 NEXT N
8100 RETURN
8200 PRINT ,,"DEMASIDO GRANDE"
8210 PRINT ,,,,"NO DEBEN SOBREPASARSE LOS 100   ELEMENTOS"
8220 FOR V=1 TO 40
8230 NEXT V
8240 CLS
8250 GOTO 7002
9500 PRINT AT 5,5;"ESTE PROGRAMA REDUCE"
9600 PRINT AT 10,10;"MATRICES"
9650 FOR T=1 TO 50
9651 NEXT T
9652 CLS
9700 RETURN
9900 SAVE "REDUCCION DE MATRICEs"
9910 RUN
```

DICTANUMEROS

COMP: TS 2068
CLAS: 'UTILITARIO
AUTOR: Julio A. Alterach



Este es un sintetizador de voz especializado en la pronunciación de los números y algunos de sus símbolos asociados.
Para la confección de este programa, se ha usado Basic, Assembler y muestreo de voz humana para generar los patrones sonoros.
El programa está compuesto de una zona de almacenamiento de datos, que se corresponde con los patrones sonoros de los números del 0 al 9, y los siguientes símbolos: "+", "–", ".", "e" (exponencial).
Para cada símbolo o número se utilizan 250 bytes, siendo éstos una digitalización de las ondas sonoras correspondientes.
La otra parte del programa, es la subrutina de sonido que toma información de la zona de datos y la convierte en una salida sonora por el parlante de la computadora y por la salida MIC (a un grabador o amplificador)
El listado de esta rutina es el siguiente:

```
SONIDO LD HL, (23758);inicio
       LD DE,250;longitud del
LOOP1  LD B,8;bloque
LOOP   LD A, (23760);demora
DEM    DEC A
       JR NZ,DEM
       BIT 0,(HL)
       JR NZ,BT1
BT0    LD A,1
       OUT (254),A;salida de
       RLC (HL);sonido bit0
       DJNZ LOOP;siguiente bit
       JR SALT
BT1    LD A,19;
       OUT (254),A;salida bit1
       RLC (HL)
       DJNZ LOOP
       JR SALT
SALT   INC HL
       DEC DE;actualización de
       LD A,D;contadores
       OR E
       JR NZ,LOOP1; terminó?
       RET
```

Este programa fue ensamblado en la dirección 65293.
En la parte basic de este programa se incluye un cargador hexadecimal, por lo que no es necesario contar con ensamblador alguno.
Los valores en las direcciones 23758 y 23759 corresponden a la dirección inicial desde donde leer los patrones sonoros.
El contenido de la memoria 23760 corresponde a la duración del bucle de retardo. Cambiando este valor mediante POKE 23760,n se modifica la velocidad de muestreo del programa assembler "SONIDO", dando como resultado una voz más grave o más aguda.

Utilización del programa

La subrutina en base propiamente dicha es la que se encuentra en la línea 9999. El resto del programa son los datos correspondientes al assembler e instrucciones de uso.
Una vez tipeado el programa, debe ejecutarse y ser grabada solamente la línea 9999, y los bytes de datos juntamente con la subrutina "SONIDO" mediante:
SAVE "dic": SAVE "dic" CODE 61666, 3800
Hay que recalcar que el primer Save se refiere a la grabación de la línea 9999 de Basic, por eso después de ejecutar el programa completo se deben borrar todas las líneas, menos la 9999 (mediante DELETE 1,9998), y luego grabar en cinta.
Para recuperarlo desde cinta teniendo otro programa en memoria, se tipea:
MERGE "": LOAD "" CODE
Recordando que la línea 9999 está reservada y no podrá usarse por el programa al cual se le quiere agregar "DICTANUMEROS".
Una de las aplicaciones más interesantes se presenta a la hora de verificar tediosos DATAs. En este caso nuestra maquinita será una grata compañía, y a cambio de un par de instrucciones similares a las que siguen, ella con gusto nos dictará el listado de los datos que hallamos tipeado.
10 READ A: LET Z$ =STR$ A: GOSUB 9999: PAUSE 20: GOTO 10

Variables útiles:

TONO: cambiando su valor en la línea 9999 obtendremos distintos tonos de voz.
Z$: string correspondiente al número a dictar.

Direcciones útiles:

65293: comienzo de la rutina "SONIDO"
23759: byte LO puntero de dirección de bloque de datos.
23758: byte HI puntero de dirección de bloque de datos
23760: temporización de la voz

```
   2 CLS : BORDER 1: OVER 0: PAP
ER 0: INK 7: RANDOMIZE 2: CLEAR
61000
   3 LET RET=60: POKE 23658,8
  10 GO SUB 1100: BEEP 1,20
  12 GO SUB 3000
  15 OVER 1
  20 GO SUB 5000: REM INSTRUCC.
  30 GO SUB 3000: STOP
  50 REM LECTURA DE DATA
  55 READ A$
  60 FOR C=1 TO LEN A$ STEP 2
  65 POKE D,16*(CODE A$(C)-48-7*
```

```
(A$(C)>"9"))+CODE A$(C+1)-48-7*(
A$(C+1)>"9")
  70 LET D=D+1
  75 NEXT C
  80 RETURN
 100 REM DATA DEL PROGRAMA DE
MANIPULACION DE SONIDO
 110 DATA "2ACE5C11FA0006083AD05
C3D20FDCB46200A3E01D3FECB0610EE1
80A3E13D3FECB0610E41800231B7AB32
0DAC9"
 200 REM LOS SIGUIENTES DATA
CORRESPONDEN A LOS PATRONES  SO-
NOROS DE CADA NUMERO O SIGNO.
  UN ERROR EN ESTOS NO DESTRUYE
EL PROGRAMA.
 309 REM ***SIGNO + ***
 310 DATA "000016810000002F0000
000009E000000001081000000239000
000003281002000060126308C62398C6
06219871199CC30"
 320 DATA "18F131891F188C6670113
133E762210D321999C440C482899D4CF
01162621917143C4CF091D117664888C
0C2898F241852C1"
 330 DATA "C2130C98F020C70202393
0E642032C4CD880C08E03580044C4899
1333222C0C4253036644488991931332
18C481993214223"
 340 DATA "4E449833000B23381364C
CE00C0A6C00181033664C98232000088
66C000000901090002408078OBD0D60E
E267C7CE1FF74B9"
 350 DATA "389D21FD1E1C6030D2C1C
80B9F32018F848E8D90AADF31E037EF7
C1A1F0F87C73CC0DFC3FC8EA7C61D1E1
3C30BC31E1D3E30"
 359 REM ***SIGNO - ***
 360 DATA "00FE000000007E8000000
00EE100000000BD00000000360100000
0027100CCC810D93365D8235244A825C
8DCA7D825B45A6D"
 370 DATA "97D24C25B74A412590359
260085004C969242104B7D2D25AAA511
25A0A50492C052101B452C08801B74A2
106922825920650"
 380 DATA "3C934805934491000000
0B4000000003401000001AD00000000
2D000000000EA000000006A005182446
ED2A36443664CC8"
 390 DATA "98411999076464CCCCCC8
C18129111A70C8CCD8CC042033331988
0817808584060061019041C0209CE000
C00E20081420172"
 400 DATA "01C000810E4030006020A
118001000E404000400 1C020002000E4
04000800041 0A2220006404400402426
00004A429290010"
 409 REM ***SIGNO . ***
 410 DATA "58210EC2CDF4C8CC2581B
C6C15610007C13D00303800F170C0F10
0E170C0F906C1E002F102C1E080F120C
18182E1C2818183"
 420 DATA "E142C1C086E186018187E
18281C087C10620800DE10630820FE00
9B00408C1008000006E000000005E000
000003601000000"
 430 DATA "160100000004E90000000
6E800000000BB00000000FC000000001
6E100000002F1000000000000000000000
000000000000000"
 440 DATA "0000000000000000000000
00000000000000000000000000000000
0369491499F42F1466400045A3E80263
F03FC4E01BE00E1"
 450 DATA "6E040E00F98CC180C13E1
22004701EC976008601FC068330C03EC
1E0104600CC140410E00E8320C406097
0000000E00C0201"
 459 REM ***SIGNO E ***
 460 DATA "4AD02E212D00FC0394044
B96A05680552A6D0591BA412DA59602D
6526A48A5084925A5B692D2486905B19
2142DA1B412D012"
 470 DATA "DA4A652F80AA496900A51
6B412A05A29F523D40B410C69250196A
0AD27B002D00E512851B400EC25A1169
422694801B413B4"
 480 DATA "84530244AC692780B4806
D25B402D02A430801A4966C05A196908
AD228412D255A446124A11E905A40680
5950A4108014C01"
 490 DATA "A4019492D00C413692FC0
050024126415A006849D3046126A49C4
17800F182DB0E410C01B421E90469069
05A007A84E804A1"
 500 DATA "06A42A413C00AA14A5160
016006C21B502E10CA51A00AC024B04A
194A5AC01B684A11A412CA59682E903D
044817403AC4BA8"
 509 REM ***NUM 0 ***
 510 DATA "0E098A131F311E32EBC38
78530C38F72C3730C0C384E1D0F873E0
4CAC130310C70C8E0C383C70C1A83663
90C18204C180870"
 520 DATA "02883540001028000004 1
A01F20401EC96E136019648B512402D0
9750A4126906DA15200902649758 3F40
A752ED34A49A325"
 530 DATA "B3006917B40A411249B60
5A590348248049128A502925A04B7C86
D06D25AA5120258002492482592 02419
640C804AC0B6592"
 540 DATA "2510250000000000000000
004202A0880025604800100001 46C18B
1980048989120003226612400081999 0
809021132010101"
 550 DATA "890CC00001600A8100201
804C000300E42310C0400818D020100C
42001C0003884C060000C02390020020
106001002810000"
 559 REM ***NUM 1 ***
 560 DATA "06FF02C10000FF00E0000
0FD00300000FE811E00003EC10681000
EF100E10002FD007800007E811E00187
2C10E011C08F900"
 570 DATA "C10601BE01E002811E611
E811C02EF32C11C00F700E108C17485E
C62113609C4262201B900000004E9000
000007B00000000"
 580 DATA "960100000016A10000000
871000000007B00000000AC000000004
E80000000168100000 01AA10016AC0CD
90744C00C999913"
 590 DATA "3127626E0011266266020
800919981C498999E846600C02281998
5E63281C086388D6030400E847018C21
E22190430861198"
 600 DATA "06300611CE000D60C5220
10410E63881020D700960100D700D600
0C21C421000301F20180010068900000
C00810081080070"
 609 REM ***NUM 2 ***
 610 DATA "D11C00380AE11E010E06F
19C812681FC220140E13E00C120E01EC
3380C3C0693DD001C02C170010681700
E8103C138068140"
 620 DATA "F00E00E02078068170000
E01E00E00868178020100E00E10C0083
80609700C1E00E10C000781F00D0082C
13C408120F00C00"
 630 DATA "C00078060130001C00C13
0001E00E11C0006017807010681 70060
100E11C008900E11E00C100780601400
01802B122000E00"
 640 DATA "ED0000281C6000000E10
00100003800C000000E0000000000F00
0000000000000004018A0A20F221A182
43E37D4212C6B1C"
 650 DATA "B66481C8A5CC446358695
B7C962C16529A1B0D85B29481128E6C0
961AD2F85A5A58B25D3686CD03990294
C1825B522D1924E"
 659 REM ***NUM 3 ***
 660 DATA "DEB2D8F05ECD599878521
869C4FB8383B1322761E0007C02E13E8
0FC8EC1FD276336A15F05DC8C4B40480
020000000800000"
 670 DATA "025013D1E5A2B12544ABD
48AA4299A05801B231E00F11E8108135
816C49CB3759A250A6D9A29A55865D82
7D825D837C83107"
 680 DATA "EC93E4BE424827D825B12
7D86D92F805DA25D34A2577887D26D14
AC82F81D848B14AE1920036886C06910
AA09804E112E516"
 690 DATA "81B600AC06D14E013F007
90A6192C03D07B144619E41B602F14EC
1B800F30AE13208D827D96E01FC82E00
E805E21F806E01E"
 700 DATA "C12E13C91EB10A812C063
602A11E6128D80520BAC743877C78361
2587838E60D1E3878E03D306278E01D2
E7F11D38FA3C970"
 709 REM ***NUM 4 ***
 710 DATA "0000000000000E9100000
E830E010006818F813E0F878E813C02E
306C11E017842E13E013C00F90E00368
3780E1932C33D06"
 720 DATA "0D60708E46118D708D623
1C5AC4231B1C18622119918C489C7841
81AC4C4460604303B73208718350991C
80E4CD824004011"
 730 DATA "0001050000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
000000000000000"
 740 DATA "00000000631F8980E680F
E20993960E18C25809E0400B13323312
2442342103144800000092402C21261
A007722B11301F0"
 750 DATA "1909C52211D9866630896
C06000089620 1C218E21CC5621D700FE
019C61D02B900021F01C000C000C1300
000000000000000"
 759 REM ***NUM 5 ***
 760 DATA "34E28FCD8F370C0D4C78E
2CD871F131727F8EE78E2BD607C5A073
260E0C28709E3873C7860F0CC5C78FE8
4CDC2977873C5B2"
 770 DATA "F3CC90914FB3399EB3842
4FEE100555B2180044DBA238415519B6
5500553DAA3A20AA037EC6C2451B45D8
411085268908112"
 780 DATA "28DAA15401000AB541002
A84EB90110A406F80900040FE8100000
07EC10000003EE10000000EF90000000
0FD00000000FE01"
 790 DATA "00000002E100000A00800
0000020000000000000000000000000001
8E46235CB621099C4990C6646109DC63
04C621181621148"
 800 DATA "8140003EFF000002FD06C
10046E90E018C02F902810002ED00C10
200BE847000C07EA13D00A05EED08000
43AFD07000826FE"
 809 REM ***NUM 6 ***
 810 DATA "840000C3864420013C91C
8B35A2DC026487933098C1065C8A7483
365C82788974D936C8A2025C827C205B
292259248B368B5"
 820 DATA "C265E8B7349065EC96486
5081A28B524A00C15DA48292145378A4
49AA39A4061202C17D00A003B0A6D450
102404949A802A4"
 830 DATA "16D09A825692F602D0164
16F01690A594A414A42D40B690A015A0
09E00F014013E006814210E01000AD00
68100169424 0100"
 840 DATA "0AD31280000AD200A0000
4ED00400004650000000A6D000000202
2C00002387E0000181131231D1085028
719F870C0E9D912"
 850 DATA "F13C5E0183C1E81884722
92AFA0239085C6A231713F13E1E1C18C
F1F0E8F283E0FCD41C6F2793C1A7860E
2C31E892C7E01C7"
 859 REM ***NUM 7 ***
 860 DATA "7468733A28304C868C198
A8D0D0708059E508A40BB8871D846411
1AA218A0DB0901161175580 2222AAE00
A0109AEA11E8108"
 870 DATA "42D602D404A128617F694
B0069097507D01AD05E4B7A03DA025C1
7D10C21B653FA002C25B25E92D2D86D9
5DA2817C8779C92"
 880 DATA "48A7DA25D2482798778AC
834088102000000000000000000000000
00000000000000000000000000000000
000000000000000"
 890 DATA "0000000000000000000000
00000000000000000000000000000000
0000000002826992770A47D68B708A00
A21CE1100005913"
```

```
 900 DATA "48259A403592493482583
A402C07B25A493582586DA13583DA482
52A45AD169A4E65B403924EA05A0065A
4695A8069246916"
 909 REM ***NUM 8 ***
 910 DATA "3E00F11C600E017806212
E817C068104E10E01E008F0068176003
C00E11CC10600F00E7002853C033906F
11E821D02798670"
 920 DATA "1DC23C42E012810EC07E0
27002606E803B7267400000000000000
00000000000000000000000000000000
0000000000000000"
 930 DATA "00000000000000000000
0000000000000000855030000B100A0
B063A12572B20F9E040CF9075B941A5F
83507224B6090AE"
 940 DATA "DFFE0277810661A171FF4
DBC5902C06D742EA728135E0F4800691
10EA5C8813324FC008448F344C01A80D
90CC1B802F11E81"
 950 DATA "E90E407000E00E40380C6
03C80F00CC00E43B904201D04B906609
FA1DE029966E17E209922ED36014610E
12E852618780D81"
 959 REM ***NUM 9 ***
 960 DATA "00003E010000000F1000
00000780000000061000000000C008C
100000E008C000000C13081000038021
100004E009C0040"
 970 DATA "00C90E406600C884C0260
06644C40C802744C899236C9367C8937
4A93500B108B35A41900372244308412
A45A91049046949"
 980 DATA "25B59A5B2865B69256809
2DA49AC4196CA69256996249A48D02C2
592019A24A9280112C82B449B44EC92B
7582306CC980364"
 990 DATA "0012102400010 0D000010
0000021000002200C800080820012000
60801009000000048168008000010D80
00000001A132424"
1000 DATA "0808234E803406D827882
926D82D922D136D1AC02792B44A412F5
2AC13D04BA5A8AB482D924A65B512CA6
7D0B48A4827D22C"
1100 REM INICIAR
1102 PRINT INK 2; P^PER 5; FLASH
 1;AT 0,4;" ????????????????????
??"
1103 PRINT INK 2; PAPER 5; FLASH
 1;AT 7,4;"?????????????????????
??"
1104 FOR I=0 TO 7 STEP 2: PRINT
INK 2; PAPER 5; FLASH 1;AT I,4;"
 ";AT I+1,4;"?"
1105 PRINT INK 2; PAPER 5; FLASH
 1;AT I,27;"?";AT I+1,27;" ": NE
XT I
1106 PRINT AT 2,7;"SINTETIZADOR
DE VOZ";AT 3,10;"PARA  NUMEROS":
 PLOT 50,135: FOR X=-30 TO -5 ST
EP .5: DRAW INK 3;3,(RND-.50)*5:
 BEEP .05,X: NEXT X
1107 GO SUB 4000: GO SUB 3000: P
RINT #0;"ESPERE 2 MINUTOS Y MEDI
O..."
1108 PAUSE 30: PRINT AT 21,13; P
APER 2; INK 6; JULIO A. ALTERA
CH"
1110 RESTORE : LET D=65293: GO S
UB 50: REM LECTURA DE PROGR.
1200 FOR L=310 TO 960 STEP 50
1203 RESTORE L
1205 LET D=((L-310)/50+240)*256+
226
1210 FOR G=1 TO 5
1250 BEEP .008,40
1260 GO SUB 50
1270 NEXT G
1280 NEXT L
1290 RETURN
3000 REM SONIDO
3010 FOR I=15 TO 0 STEP -1: FOR
J=10+3*I TO I*3 STEP -3: BEEP .0
1,J: NEXT J: NEXT I
3020 FOR I=0 TO -30 STEP -1: BEE
P .01,I: NEXT I
3030 RETURN
4000 REM GRAFICO
4010 FOR I=106 TO 0 STEP -5
4020 INK I/15+1: LET L=175-I: LE
T X=27/106*I
4030 PLOT X,175: DRAW 0,-L: DRAW
 256-2*X,0: DRAW 0,L
4040 NEXT I
4050 INK 7
4060 FOR X=0 TO 255 STEP 255/20
4070 PLOT INK 8;X,0: DRAW INK 8;
27*(125-X)/125,106
4080 NEXT X
4100 RETURN
5000 REM INSTRUCCIONES
5010 CLS
5020 LET A$="Este programa logra
 sintetizar  una voz digital, pr
onunciando  numeros y ciertos s
ignos contenidos en la variable
Z$.": GO SUB 8000
5025 GO SUB 8500
5030 CLS : LET A$="La forma de u
sarlo es simple.   Por ejemplo,
si quiere que la   maquina pronu
ncie el numero uno,debe asignar
'1' a Z$ y llamar  a la subrutin
a de un solo paso  en la linea 9
999.": GO SUB 8000
5032 GO SUB 8500
5035 PRINT AT 10,0: LET A$=
  "    LET A$=""1"": GO SUB 999
9                    suena....":
GO SUB 8000: PAUSE RET: LET z$="
1": GO SUB 9999
5040 GO SUB 8500
5050 CLS : LET A$="Otro ejemplo
usando mas de una  cifra:"+CHR$
13+"    LET A$=""4490"": GO SUB 9
999          suena....": GO SUB 800
0: PAUSE RET: LET z$="4490": GO
SUB 9999
5060 GO SUB 8500
5070 CLS : LET A$="Estan incluid
os los signos rela-cionados con
numeros.                Estos son :":
 GO SUB 8000
5080 PAUSE 30: PRINT : LET a$=".
(punto)": LET z$=".": GO SUB 8
000: GO SUB 9999
5090 PAUSE 30: PRINT : LET a$="-
 (menos)": LET z$="-": GO SUB 8
000: GO SUB 9999
5095 PAUSE 30: PRINT : LET a$="+
 (mas)": LET z$="+": GO SUB 800
0: GO SUB 9999
5100 PAUSE 30: PRINT : LET a$="E
 (exponencial)": LET z$="E": GO
 SUB 8000: GO SUB 9999
5110 GO SUB 8500
5120 CLS : LET A$="         ....EJE
MPLOS....": GO SUB 8000
5130 PRINT '': LET A$="LET Z$=""
3.14"":GOSUB 9999": GO SUB 8000:
 PAUSE RET: LET Z$="3.14": GO SU
B 9999: GO SUB 8500
5140 PRINT '': LET A$="LET Z$=""
-8.51"":GOSUB 9999": LET Z$="-8.
51": GO SUB 8000: PAUSE RET: GO
SUB 9999: GO SUB 8500
5150 PRINT '': LET A$="LET Z$=""
2.45E+12"":GOSUB 9999": LET Z$="
2.45E+12": GO SUB 8000: PAUSE RE
T: GO SUB 9999: GO SUB 8500
5160 PRINT '': LET A$="Utilizand
o STR$ ......": GO SUB 8000: PAU
SE 40
5170 PRINT '': LET A$="LET FF=23
:LET Z$=STR$ FF:GOSUB 9999": GO
SUB 8000: PAUSE RET: LET FF=23:
LET Z$=STR$ FF: GO SUB 9999: GO
SUB 8500
5180 PRINT '': LET A$="LET PEPE=
4*5: LET Z$=STR$ PEPE:GOSUB 9999
": GO SUB 8000: PAUSE RET: LET P
EPE=4*5: LET Z$=STR$ PEPE: GO SU
B 9999: GO SUB 8500
5190 CLS : LET A$="El programa e
s util en el momen-to de verific
ar una serie de va-lores introdu
cidos en la maquinay simplemente
, a medida que la   computadora l
os va dictando se   comparan con
los  numeros ori-  ginales.": GO
 SUB 8000: GO SUB 8500
5200 PRINT '': LET A$="Esto ulti
mo es aplicable a la verificacio
n de sentencias DATA    mediante
un bucle:                  10 READ
 A:LET Z$=STR$ A:GOSUB 9999:PAUS
E 10: GOTO 10"
5210 GO SUB 8000: GO SUB 8500
5220 CLS : LET A$="Se puede usar
 el programa para  dictar numero
s en el caso de no poseer una im
presora.": GO SUB 8000: GO SUB 8
500
5225 CLS : LET a$="Modificando e
l valor a la varia-ble reservada
 TONO en la subrutina de la line
a 9999, obtendra uncambio en la
pronunciacion.": GO SUB 8000: GO
 SUB 8500
5230 CLS : LET A$="Para guardar
en cinta el progra-ma :"+CHR$ 13
+"1-borrar todas las lineas meno
s la 9999.": GO SUB 8000: GO SUB
 8500
5240 PRINT : LET A$=              "
2-guardar en cinta (SAVE ""voz""
)": GO SUB 8000: GO SUB 8500
5250 PRINT : LET a$=              "
3-hacer :"+CHR$ 13+"     SAVE ""v
oz"" CODE 61600,3760": GO SUB 80
00: GO SUB 8500
5300 CLS : LET A$=                "
Para cargar el programa teniendo
otro en memoria, usar:
MERGE """":LOAD """" CODE
     recordando que la linea 9999
     esta reservada.": GO SUB 800
0: GO SUB 8500
5305 PRINT '': LET a$=            "
Las variables Z, Z$ y TONO estan
reservadas para uso de la subru-
tina.": GO SUB 8000: GO SUB 8500
5307 CLS : LET A$="SIGNOS Y NUME
ROS PERMITIDOS:"+CHR$ 13+"0"+CHR
$ 13+"1"+CHR$ 13+"2"+CHR$ 13+"3"
+CHR$ 13+"4"+CHR$ 13+"5"+CHR$ 13
+"6"+CHR$ 13+"7"+CHR$ 13+"8"+CHR
$ 13+"9"+CHR$ 13+"+"+CHR$ 13+"-"
+CHR$ 13+"."+CHR$ 13+"E":
5308 GO SUB 8000: GO SUB 8500
5310 CLS : LET a$="quiere ver nu
evamente las ins-  trucciones ?
(s/n)": GO SUB 8000
5320 INPUT LINE a$: IF a$="S" TH
EN GO TO 5000
5330 RETURN
8000 REM VISUALIZACION
8005 OVER 0: PLOT 0,0: DRAW 255,
0: DRAW 0,175: DRAW -255,0: DRAW
 0,-175: OVER 1
8010 FOR I=1 TO LEN A$
8020 PRINT A$(I);
8025 IF I/3=INT (I/3) THEN BEEP
.005,RND*40
8030 NEXT I
8040 RETURN
8500 REM RETARDO
8510 PAUSE 40: PRINT #0;AT 0,0;"
presione una tecla para seguir..
"
8520 IF INKEY$="" THEN GO TO 852
0
8525 PRINT #0;AT 0,0;"
                                "
8530 RETURN
9999 LET TONO=28: POKE 23758,226
: FOR Z=1 TO LEN z$: POKE 23760,
TONO+3*(Z=LEN Z$): POKE 23759,((
CODE z$(Z)+196) AND CODE z$(Z)>=
45)-(22 AND CODE Z$(Z)=69)+(240
AND CODE Z$(Z)=43): RANDOMIZE US
R 65293: NEXT Z: RETURN
```

MANEJO DE INTERRUPCIONES EN EL Z-80

IMØ

IM1

IM2

Junto con las centenares de instrucciones assembler del Micro Procesador Z-80 conviven tres de ellas que merecen un capítulo aparte. Por eso veremos detalladamente cómo se lucen éstas en el dominio de las interrupciones del chip.

El Programador de Código de Máquina se alista a transmitir sus primeros logros a la expectante criatura electrónica.

Después de una ímproba tarea observa, ya sin asombrarse, que sus ilusiones se atomizan en cuestión de segundos. Desalentado y con rabia, el pequeño Homo Sapiens murmuró:

-Otra vez colgada... Morite!

El clic del interruptor desconectó los signos vitales del microprocesador. El compucidio se hizo patente un segundo más tarde y sin ceremonias ulteriores, la computadora fue enterrada con su ataúd de telgopor en una fosa acartonada hecha a medida. Esta fue una escena corriente durante unas semanas, hasta que el espécimen advirtió que el citado fenómeno ocurría al manejar la siguiente instrucción, mientras intentaba hacer correr más rápidamente sus programas, conforme a un consejo de su tío Mono:

DI

Finalmente, como nuestro Homo Sapiens tiene vocación de servicio, ha decidido comunicarnos sus hallazgos, como aporte al sindicato de computadoras colgadas.

Para comenzar, una interrupción será justamente éso, una discontinuidad. Posteriormente veremos cuál fue el problema de nuestro asesino de transistores.

El Z-80 tiene 3 maneras de responder frente a una interrupción, canalizadas en 3 modos: IMO (Interrupt Mode 0), IM1, e IM2.

Por omisión se sobreentiende IM0 (es decir, desde que se le da tensión a procesador, hasta que se le ordena cambiarlo). Esto le permite responder frente a una de ellas de igual manera que su padre, el INTEL 8080; uno de los requisitos para que el Z-80 sea compatible con todo el software de su papá, por ejemplo CP/M. Este tipo de interrupciones no interesaron al Homo Sapiens porque suponía un nivel demasiado alto de programación y porque para sus necesidades no aportaba nada.

En el IM1 (que usa nuestra computa-

dora), cuando se produce una interrupción, el control del programa pasa a la dirección 38Hex, con un proceso similar al del RST. Allí se encuentran las rutinas que leen el teclado e incrementan la variable del sistema FRAMES.

Este proceso se realiza con una frecuencia de 62.5 veces por segundo de la manera siguiente: por una de las líneas (patitas) del procesador, se le solicita a éste la interrupción mediante una señal. Esté donde esté corriendo, el procesador responderá de acuerdo con el modo en que se encuentre. En IM1 llamará a 38Hex y después volverá al lugar en el que estaba realizando las tareas cuando lo "pescó" la interrupción, siguiendo como si nada lo hubiese molestado.

La implementación de tal artificio asombró al Sapiens ya que con ésto nos evitábamos armar un sistema operativo que hubiera tenido que leer el teclado de una manera más engorrosa.

Pero está de más decir que no todos los programas requieren que cada 1/60s se lea el teclado. Ni siquiera el pulpo extraterrestre que tiene nuestro personaje encerrado en la pecera podría pasarse más de un minuto apretando teclas sin parar, para exprimir a fondo semejante capacidad. Su programa, que calculaba integrales, requería una única entrada de datos al principio (léase necesidad de interrupciones que "sientan" el exterior), de modo que las 60 veces que le leían el teclado mientras calculaba, eran desperdiciadas.

En eso andaba Sapiens cuando colgó su máquina. Sabía que si le tapaba las orejas al Z-80 (cortando las interrupciones mientras calculaba), éste no escucharía la señal del chip periférico que le recordaba su interrupción, por lo que ganaría valioso tiempo al no ejecutarse la rutina 38Hex. Y tan bien lo hizo que el procesador quedó sordo durante lo que restó de su periódica vida (porque después de haberlas cortado y corrido, no fueron habilitadas). La única salida fue empezar todo de nuevo, vía enchufe o interruptor.

Las instrucciones que controlan la habilitación de las interrupciones son EI (enable interrupt) y DI (disable interrupt). Después de un EI, el flag IFF (Interrupt Flip Flop) es prendido, permitiendo a la señal de interrupciones ser reconocida. Este flag permanecerá así hasta que un DI se ejecute (lo que hará obviarlas) o nos encontraremos en una rutina de interrupción. Esto último, porque si al encontrarnos allí, el procesador responde a la interrupción que llega, saltará nuevamente a la rutina que está corriendo, re-llamándose hasta el infinito.

De modo que antes de volver desde la rutina de interrupciones con RETI (RET desde una interrupción), deberemos poner EI (para habilitarlas nuevamente). Una espera en el reconocimiento de las interrupciones después de ejecutarse dicha instrucción permite fijarse si lo que sigue es un RETI, y de esa manera saltar enseguida al programa principal, retomándolo, sin peligro de que las interrupciones nos "atrapen" justo cuando salimos de la rutina a la cual nos habían mandado. Como la desinterruptitis es contagiosa e histerizante, Sapiens aconsejó usar estas instrucciones de a pares, nunca de a una por vez, sobre todo DI.

El tercer modo de interrupción está pensado para el uso del pueblo; y para facilitarle las cosas al pueblo, todo lo comunicado puede aplicarse. Al seleccionar el IM2, la rutina 38Hex dejará de realizarse, porque pertenece a una modalidad que ya no corre. En este universo podemos ordenarle al procesador que salte a la rutina de interrupciones que querramos.

Cuando se le solicita la interrupción en IM2, por el bus de datos se la envía al procesador 8 bits que corresponden al byte menos significativo de una dirección a la que llamaremos puntero. El byte más significativo de esa dirección es aportado por el registro I. Esta dirección puntero (resultante de 1*256 +bus) contiene otra dirección que indica el lugar al cual saltará en la interrupción. (Ej.: si I =128 y bus =10, la dirección puntero será 32768, y en 32768 y 32769 se encontrará guardada la verdadera localización de la rutina de interrupciones).

Aquí nos encontramos con el conflicto que ocasionó unas largas vacaciones en el Borda a nuestro Sapiens. Le resultó imposible averiguar cuál era el dato que mandaban al bus, por lo que la dirección puntero podía variar en un rango de 256 posibilidades. Por ej.: si I =128 y el bus de datos es desconocido, la dirección puntero puede variar entre 32768 y 33024. Si nosotros supiéramos que el dato del bus es cero, estaríamos seguros de que la dirección puntero se encontraría en 32768 y 32769. Pero ni nosotros sabemos eso, ni el Sapiens lo supo. Su psiquiatra, una medusa tropical, le aconsejó que se fabricara una "pista de atruuzaje", que consistía en un bloque íntegramente cargado con 254, desde 32768 hasta 33025 (es-

to para I =128; si I =133, desde 34048 hasta 34305, etc.). Cualquiera fuese el dato del bus, combinado con el registro I indicaría una dirección de esa "pista" y como el único dato que existe allí es 254, la dirección puntero y la siguiente contendrán 254. De esa manera se obtendrá la dirección 65278 (254*265 +254), que es a la cual teóricamente debería saltar el procesador en cada interrupción. Si quisiéramos saltar a otra dirección, ponemos en la pista otro dato, por ejemplo 200 y calculamos la dirección a la cual saltará: 200*256 +200. Esta dirección puede ser también un JUMP a otra dirección que contenga realmente la rutina en cuestión.

Este tipo de rutinas se manejan de la manera habitual; pueden tener CALLS y RETS para llamar a otras direcciones útiles. Cuando encuentren un RETI (si los CALLS y RETS fueron usados convenientemente) volverán al programa principal, dando por terminada la rutina de interrupciones.

Dos últimas recomendaciones. Las rutinas deberán ser cortas, porque al realizarse constantemente retardan mucho al programa. Calculen el tiempo de la rutina y multiplíquenlo por 62.5 para tener el incremento de tiempo para 1 segundo. Lo que sin interrupciones se realizaba en 1 segundo, ahora tardará 1 segundo más ese tiempo calculado, porque cada 0.016s se nos "mete" una interrupción en el medio, como una cuña. La otra es hacer PUSH y POP con todos los registros (al principio de la rutina y al final, respectivamente) ya que si bien las interrupciones se manejan independientemente del programa, usan sus registros y flags, ya que, después de todo, también son programas.

Las interrupciones pueden servir para poner música de fondo a nuestros juegos independientemente del programa principal, para temporizar programas científicos y recibir información en controles de procesos a un ritmo determinado (temperatura, humedad, presión, etc.), para leer la posición de algún joystick, mouse o lápiz óptico constantemente. En fin, cualquier tarea que requiera ser realizada rítmicamente y en forma independiente del programa principal, podrá ser optimizada con el uso de las interrupciones.

Antes de despedirnos conviene saber que aparte de las interrupciones que se controlan por software con DI y EI, llamadas enmascarables, existen otras que no lo son. Una señal diferente es usada para realizar este tipo de interrupciones que siempre ocasionan un salto a 56Hex. A diferencia de sus parientes, no se realizan constantemente; sólo entran en el escenario cuando algún periférico se lo pide (por una de las salidas que tiene la computadora atrás).

Programa 1:

(no lleva programa en BASIC)

```
DI
RET
```

Si quisiéramos cortar las interrupciones en algún momento, podríamos usar esta rutina o simplemente hacer DI en el medio del programa, a partir de donde querramos que se corten, tal como lo hizo nuestro personaje. Pero cuando se requiera el servicio de las interrupciones (en IM1 la lectura del teclado y en IM2 nuestra rutina) para mantener el control del sistema, la máquina estará colgada (En IM1, por ej., no se leerá el teclado). Eso sí, todo lo que corra lo hará más rápido.

Programa 2:

(no lleva programa en BASIC)

```
DI
RUTINA A
EI
```

Esto sería lo correcto, siempre y cuando la rutina A no requiera el uso de las interrupciones. Antes de realizarse, DI le "tapa las orejas" al Z-80 para que corra esta rutina más rápido (léase sin interrupciones ni molestias del exterior). Al final, con EI, retornamos al mundo normal.

Los programas anteriores pueden correrse con esos resultados en cualquiera de los modos de interrupción. Al cortar las interrupciones, no se llamará más a la rutina correspondiente, acarreando los efectos de no realizarse la función que tenía confiada.

Programa 3:

(lleva programa en BASIC)

-Bloque cargado con 250 entre 32768 y 33025

```
-En 3000:
DI
PUSH AF
LD A,128
LD I,A
POP AF
IM2
EI
RET
-En 64250:
PUSH AF
PUSH HL
LD HL,65535
A1 LD A, (HL)
OUT (254),A
JR NZ,A1
POP HL
POP AF
EI
RETI
```

La máquina se cuelga, esta vez con las interrupciones puestas, en IM2 y haciendo la rutina que le ordenamos. Vale la pena recordar los push y pop antes y después de la rutina para no alterar los registros, así como usar los CALL y RET de a pares para no cambiar el lugar del stack a la vuelta de cada interrupción. No necesariamente nos limitaremos a hacer una rutina lineal que no salta a ningún lado, sino que podemos llamar a cualquier dirección sin ningún temor, pero estando seguros de que volveremos a la la rutina al programa principal con el stack idéntico al que teníamos antes de la interrupción (salvo si queremos cambiarlo intencionalmente porque implementamos tal uso).

```
10 FOR a=32768 TO 33025
20 POKE a,250
30 NEXT a
40 DATA 243,245,62,128,237,71,241,237,94,251,201
50 DATA 245,229,33,255,255,126,211,254,53,32,250,225,241,251,237,77
60 FOR a=30000 TO 30010
70 READ c
80 POKE a,c
85 NEXT a
90 FOR a=64250 TO 64265
100 READ c
110 POKE a,c
120 NEXT a
130 PRINT "PRESIONE ENTER PARA VER-         LOS PROGRAMAS QUE RESTAN ESTAN  DEL OTRO LADO": PAUSE 0
140 RANDOMIZE USR 30000
```

Programa 4:

(lleva programa en BASIC)
-Igual a 3 pero en 64250 en lugar de RETI poner JP 38Hex

Acá conseguimos el mismo efecto que antes, pero sin colgarnos, porque saltamos a 38Hex. Le agregamos al principio de 38Hex ese pedacito que hicimos nosotros. Más exactamente, 38Hex es una continuación de nuestra rutina (no es un call que llama a una sub-rutina y después vuelve, es un JUMP que ordena seguir con lo que encuentre allí). Para nuestra tranquilidad, al final de 38Hex un pequeño arreglo burocrático aún desconocido devuelve el control al programa principal, volviendo de la interrupción como si encontrase EI RETI.

```
  10 FOR a=32768 TO 33025
  20 POKE a,250
  30 NEXT a
  40 DATA 243,245,62,128,237,71,
241,237,94,251,201
  50 DATA 245,229,33,255,255,126
,211,254,53,32,250,225,241,195,5
6,00
  60 FOR a=30000 TO 30010
  70 READ c
  80 POKE a,c
  85 NEXT a
  90 FOR a=64250 TO 64265
 100 READ c
 110 POKE a,c
 120 NEXT a
 130 PRINT "PRESIONE ENTER PARA
VER": PAUSE 0
 140 RANDOMIZE USR 30000
```

Programa 5:

(lleva programa en BASIC)
-Igual a 3, pero en 64250 en lugar de la rutina va JP 38Hex y nada más.

Obtenemos una simulación de IM1 en IM2. Se invoca a 64250 y desde allí se salta (no se llama) a 38Hex. El efecto es el mismo que el que nuestra computadora realiza en IM1.

```
  10 FOR a=32768 TO 33025
  20 POKE a,250
  30 NEXT a
  40 DATA 243,245,62,128,237,71
241,237,94,251,201
  50 DATA 195,56,00
  60 FOR a=30000 TO 30010
  70 READ c
  80 POKE a,c
  85 NEXT a
  90 FOR a=64250 TO 64252
 100 READ c
 110 POKE a,c
 120 NEXT a
 130 PRINT "PRESIONE ENTER PARA
VER": PAUSE 0
 140 RANDOMIZE USR 30000
```

Programa 6:

(lleva programa en BASIC)
-Bloque entre 32768 y 33025 con 250
-En 30000 la rutina que va en esa dirección de 3.
-En 64250:

```
PUSH AF
PUSH HL
JP 40000
```

-En 40000:

```
LD HL,50000
LD (64253),HL
LD A,O
OUT (254),A
POP HL
POP AF
JP 38HEX
```

-En 50000:

```
LD HL,400000
LD (64253),HL
LD A,7
OUT (254),A
POP HL
POP AF
JP 38HEX
```

Las cosas se ponen más interesantes. En la primera interrupción, después de habilitar el IM2, saltamos a 40000. Allí se cambia la dirección del JUMP a 50000 y saltamos a 38Hex para mantener el control de la máquina. En la segunda, saltamos a 50000 (recordemos que el JUMP había sido cambiado en la interrupción anterior). Cambiamos el JUMP nuevamente a 40000 y saltamos a 38Hex. En la tercera, saltamos a 40000, etc... El efecto se consigue a partir de dos posibles derivaciones, una a 40000 y otra a 50000, con 2 datos distintos a mandar por el port 254. La implementación cuidadosa de este tipo de estructuras, junto a programas que se automodifican pueden llegar a ser herramientas de gran poder tanto para el programador como para quien quiera ejercitarse en razonamientos complicados.

```
  10 FOR a=32768 TO 33025
  20 POKE a,250
  30 NEXT a
  40 DATA 243,245,62,128,237,71
241,237,94,251,201
  50 DATA 245,229,195,64,156
  52 DATA 33,80,195,34,253,250,6
2,0,211,254,225,241,195,56,00
  54 DATA 33,64,156,34,253,250,6
2,7,211,254,225,241,195,56,00
  60 FOR a=30000 TO 30010
  70 READ c
  80 POKE a,c
  85 NEXT a
  90 FOR a=64250 TO 64254
 100 READ c
 110 POKE a,c
 120 NEXT a
 130 FOR a=40000 TO 40014
 140 READ c
 150 POKE a,c
 160 NEXT a
 170 FOR a=50000 TO 50014
 180 READ c
 190 POKE a,c
 200 NEXT a
 210 PRINT "PRESIONE ENTER PARA
VER": PAUSE 0
 220 RANDOMIZE USR 30000
```

Programa 7:

(no lleva programa en BASIC)

```
DI
HALT
```

La instrucción HALT detiene al procesador hasta que se realice una interrupción. Cuando la señal llega, realiza la rutina de interrupciones y continúa el programa de manera normal. Como DI corta las interrupciones, HALT se queda esperando algo que nunca llegará.

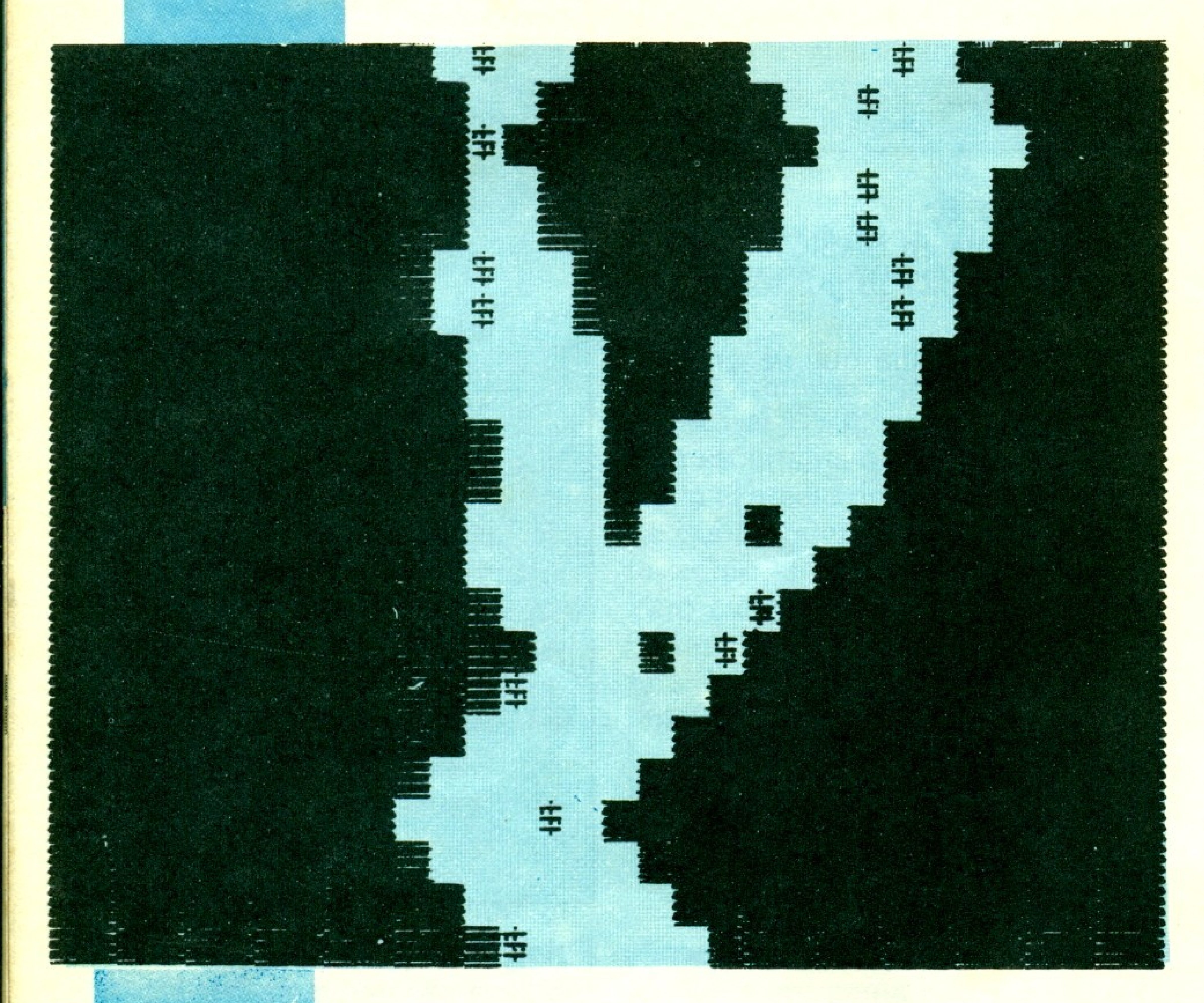
FIN

programa principal, podrá ser optimizada con el uso de las interrupciones.

Antes de despedirnos conviene saber que aparte de las interrupciones que se controlan por software con DI y EI, llamadas enmascarables, existen otras que no lo son. Una señal diferente es usada para realizar este tipo de interrupciones que siempre ocasionan un salto a 56Hex. A diferencia de sus parientes, no se realizan constantemente; sólo entran en el escenario cuando algún periférico se lo pide (por una de las salidas que tiene la computadora atrás).

FORMULA 1

COMP.: TK 83/85; CZ 1000/1500
CONF.: 16 K
CLAS.: JUEGO
AUTOR: Pablo Ignacio Bazán



Comentarios previos

Comienza la carrera con una presentación del juego donde se explica su objetivo, cómo se maneja el auto y la forma de conteo de puntos.
Luego, presionando una tecla, el juego comienza.
Aparecen la pista y el auto titilando en la parte media inferior de la pantalla. Es el gráfico SHIFT A.
Este se mueve a izquierda y derecha por medio de las teclas 5 y 8 respectivamente.
El objetivo es lograr permanecer el mayor tiempo posible en la pista, sumando puntos.

```
POKEAR LO SIGUIENTE DESDE LA
POSICION 16514.
RECUERDE QUE DEBE RESERVAR 13
BYTES EN LA LINEA 1:
1 REM 00...(140 CEROS)..00

205 146 64  177 195 186 64  205
146 64  47  177 47  195 186 64
62  3   203 57  48  1   7   205
56  48  2   7   7   38  0   104
229 6   5   41  16  253 209 25
9   79  237 91  12  64  25  35
126 203 127 40  2   238 143 201
203 95  40  2   238 143 119 201
33  254 31  126 43  230 127 79
126 230 63  254 47  56  2   214
32  128 71  203 25  229 56  5
205 130 64  24  3   205 137 64
225 124 183 32  222 201 205 28
28  1   214 2   42  12  64  9
84  93  1   181 2   42  12  64
9   237 184 17  32  0   25  54
128 43  29  194 1   65  32  250
201
```

Una forma de lograr mayor puntaje es pisar todos los "$" que encontremos en nuestro camino, pues cada uno de éstos vale 10 puntos.

La carrera consta de cinco vueltas al circuito, que al terminar nos darán 75 puntos. De no haber chocado obtendrá 150 puntitos más; si chocó un auto tendrá sólo 100 puntos fijados por la computadora.

Si tiene la suerte y la habilidad de acceder al podio, su ordenador le pedirá sus iniciales, a lo que deberá responder con los siguientes comandos:

K: pasa las letras en el orden del abecedario.

J: pasa las letras en sentido inverso al del abecedario.

O: fija la letra vista en pantalla y pasa a la siguiente.

S: ubica un espacio en el cuadro y pasa al siguiente.
Es un buen entretenimiento que se presta para jugar con los amigos, y por qué no, para hacer pasar un papelón a los no tanto.

Variables importantes

M(): matriz de los puntos de records.
M$(): matriz de iniciales de records.
B$(), A$(): matrices que contienen la pista.
L: cantidad de choques.
P: cantidad de puntos ganados.
C: número de vuelta en la que se chocó.

Estructura del programa

2-100: inicialización de variables y registros.
104-600: desarrollo de la lógica principal.
700-7600: distribución de puntaje, información sobre su carrera y puntaje, y lógica de acceso a la tabla de records.
8000-8100: inicialización de las variables que contienen la pista.
9800-9998: rutina de explicación de uso del juego.

```
   2 GOSUB 9800
   3 DIM M(6)
   4 DIM M$(6,3)
   5 DIM B$(15,31)
   6 DIM A$(15,31)
  10 FOR F=1 TO 5
  11 LET M(F)=100
  12 LET M$(F)="TS "
  13 NEXT F
  49 GOSUB 8000
  50 LET S=0
  52 LET Q=708+PEEK 16396+256*PE
EK 16397
  54 LET X=0
  55 LET L=0
  56 LET C=0
  58 LET K=0
  95 FOR H=0 TO 21
  96 PRINT AT H,0;"[redacted]"
  97 NEXT H
 100 POKE Q,8
 104 FOR J=1 TO 5
 105 FOR G=1 TO 11
 106 LET U$="125"
 120 PRINT AT USR 16619,0;A$(G)
 121 LET X=X+(INKEY$="8")-(INKEY
$="5")
 122 IF PEEK (Q+X)=13 THEN LET S
=S+1
 123 IF PEEK (Q+X)=128 THEN GOTO
 2000
 125 POKE Q+X,8
 130 NEXT G
 140 FOR G=1 TO 12 STEP .5
 145 LET U$="165"
 160 PRINT AT USR 16619,0;A$(INT
 (G))
 161 LET X=X+(INKEY$="8")-(INKEY
$="5")
 162 IF PEEK (Q+X)=13 THEN LET S
=S+1
 163 IF PEEK (Q+X)=128 THEN GOTO
 2000
 165 POKE Q+X,8
 170 NEXT G
 180 FOR G=1 TO 4 STEP .5
 185 LET U$="205"
 200 PRINT AT USR 16619,0;B$(INT
 (G))
 201 LET X=X+(INKEY$="8")-(INKEY
$="5")
 202 IF PEEK (Q+X)=13 THEN LET S
=S+1
```

```
 203 IF PEEK (Q+X)=128 THEN GOTO
 2000
 205 POKE Q+X,8
 210 NEXT G
 220 FOR G=4 TO 8
 225 LET U$="245"
 240 PRINT AT USR 16619,0;B$(G)
 241 LET X=X+(INKEY$="8")-(INKEY
$="5")
 242 IF PEEK (Q+X)=13 THEN LET S
=S+1
 243 IF PEEK (Q+X)=128 THEN GOTO
 2000
 245 POKE Q+X,8
 250 NEXT G
 260 FOR G=9 TO 12 STEP .5
 265 LET U$="285"
 280 PRINT AT USR 16619,0;B$(INT
 (G))
 281 LET X=X+(INKEY$="8")-(INKEY
$="5")
 282 IF PEEK (Q+X)=13 THEN LET S
=S+1
 283 IF PEEK (Q+X)=128 THEN GOTO
 2000
 285 POKE Q+X,8
 290 NEXT G
 300 FOR G=12 TO 15
 305 LET U$="325"
 320 PRINT AT USR 16619,0;B$(G)
 321 LET X=X+(INKEY$="8")-(INKEY
$="5")
 322 IF PEEK (Q+X)=13 THEN LET S
=S+1
 323 IF PEEK (Q+X)=128 THEN GOTO
 2000
 325 POKE Q+X,8
 330 NEXT G
 350 LET C=C+1
 400 NEXT J
 499 FOR H=0 TO 21
 500 LET U$="505"
 501 LET NADA=USR 16619
 502 LET X=X+(INKEY$="8")-(INKEY
$="5")
 503 IF PEEK (Q+X)=13 THEN LET S
=S+1
 504 IF PEEK (Q+X)=128 THEN GOTO
 2000
 505 POKE Q+X,8
 506 IF H=6 THEN PRINT AT 0,0;"[redacted]"
 507 IF H=6 THEN GOTO 520
 510 PRINT AT 0,0;"[redacted]"
 530 NEXT H
 560 FOR W=1 TO 70
 570 NEXT W
 600 CLS
 700 PRINT "MUY BIEN, TERMINASTE
 EL CIRCUITO",,,,,,,
 750 PRINT "POR ESTO TENES 75 PU
NTOS EXTRAS"
 755 FOR Y=1 TO 25
 756 NEXT Y
 800 CLS
 801 IF L=0 THEN PRINT AT 5,0;"T
E QUEDARON LOS TRES AUTOS SANOS"
 802 IF L=1 THEN PRINT AT 5,0;"T
E QUEDARON DOS AUTOS SANOS"
 803 IF L=2 THEN PRINT AT 5,0;"T
E QUEDO UN AUTO SANO"
 805 PRINT AT 10,0;"POR ESO TENE
S ";(3-L)*50;" PUNTOS MAS"
 810 LET K=75
 812 FOR Y=1 TO 15
 814 NEXT Y
 820 FOR P=1 TO 20
 830 NEXT P
 850 LET P=K+S*10+C*10+(3-L)*50
 860 IF P=0 THEN GOTO 910
 880 LET P=P+INT (RND*9)
 910 CLS
 920 PRINT "TUS PUNTOS SON:";P
 950 GOTO 6000
2000 FOR F=1 TO 4
2100 POKE Q+X,8
2110 POKE Q+X,0
2120 NEXT F
2121 POKE Q+X,8
2122 LET L=L+1
2123 IF L=3 THEN LET NADA=USR 16
578
2124 IF L=3 THEN GOTO 2150
2125 POKE Q+X,128
2126 IF PEEK (Q+X+1)=128 THEN LE
T X=X-1
2127 IF PEEK (Q+X-1)=128 THEN LE
T X=X+1
2128 POKE Q+X,8
2129 FOR U=1 TO 8
2130 NEXT U
2131 GOTO VAL U$
2150 FOR U=1 TO 8
2155 NEXT U
2200 CLS
2300 PRINT "CHOCASTE ",,
2400 PRINT "EN LA VUELTA NUMERO
";C
2500 GOTO 820
4000 DIM R$(3,1)
4002 LET D=342+PEEK 16396+256*PE
EK 16397
4004 PRINT AT 8,9;"[redacted]"
4006 PRINT AT 9,9;"[redacted]   [redacted]   [redacted]   [redacted]"
4008 PRINT AT 10,9;"[redacted]   [redacted]   [redacted]
[redacted]"
4010 PRINT AT 11,9;"[redacted]   [redacted]   [redacted]
[redacted]"
4012 PRINT AT 12,9;"[redacted]"
4020 FOR F=0 TO 8 STEP 4
4025 LET G=38
4030 POKE D+F,G
4040 IF INKEY$="" THEN GOTO 4040
4050 LET K$=INKEY$
4070 IF K$="K" THEN LET G=G+1
4080 IF K$="J" THEN LET G=G-1
4090 IF K$="0" THEN GOTO 4200
4100 IF G<38 THEN LET G=63
4110 IF G>63 THEN LET G=38
4115 IF K$="S" THEN LET G=0
4120 POKE D+F,G
4130 IF K$="S" THEN GOTO 4200
4140 GOTO 4040
4200 LET R$((F/4)+1)=CHR$ G
4235 IF INKEY$<>"" THEN GOTO 423
5
4240 NEXT F
4260 LET M$(6)=R$(1)+R$(2)+R$(3)
4265 CLS
4270 GOTO 6061
6003 FOR T=1 TO 5
6004 IF P>M(T) THEN GOTO 4000
6005 NEXT T
6006 GOTO 7000
6062 FAST
6065 LET M(6)=P
6070 FOR F=1 TO 6
6080 FOR G=1+F TO 6
6090 IF M(F)>=M(G) THEN GOTO 614
0
6100 LET O=M(F)
6105 LET M(F)=M(G)
6110 LET M(G)=O
6120 LET F$=M$(F)
6125 LET M$(F)=M$(G)
6130 LET M$(G)=F$
6140 NEXT G
6150 NEXT F
6160 SLOW
7000 CLS
7010 PRINT "LOS MEJORES RECORDS
SON :"
7020 FOR F=1 TO 5
7030 IF M(F)>=1000 THEN PRINT AT
 3+(F*2),9;M$(F);"....";M(F)
7035 IF M(F)<1000 THEN PRINT AT
3+(F*2),9;M$(F);".....";M(F)
7040 NEXT F
7045 PRINT AT 21,0;"TOQUE UNA TE
CLA PARA JUGAR"
7050 IF INKEY$="" THEN GOTO 7050
7060 CLS
7600 GOTO 40
8000 LET B$(1)="[redacted]$
[redacted]"
8001 LET B$(2)="[redacted]
[redacted]"
8002 LET B$(3)="[redacted]   [redacted]   [redacted]
[redacted]"
8003 LET B$(4)="[redacted]   [redacted]
[redacted]"
8004 LET B$(5)="[redacted]   [redacted]
[redacted]"
8005 LET B$(6)="[redacted]   $   [redacted]
[redacted]  $ [redacted]"
8006 LET B$(7)="[redacted]   $   [redacted]
[redacted]"
8007 LET B$(8)="[redacted]   [redacted]
[redacted]$[redacted]"
8008 LET B$(9)=B$(7)
8009 LET B$(10)=B$(6)
8020 LET B$(11)=B$(5)
8021 LET B$(12)=B$(4)
8022 LET B$(13)=B$(3)
8023 LET B$(14)=B$(2)
8024 LET B$(15)=B$(1)
8080 LET A$(1)="[redacted]$ [redacted]
[redacted]"
8081 LET A$(2)="[redacted]
 $[redacted]"
8082 LET A$(3)="[redacted]
[redacted]"
8083 LET A$(4)="[redacted]
[redacted]"
8084 LET A$(5)="[redacted]
 $    [redacted]"
8085 LET A$(6)=A$(4)
8086 LET A$(7)=A$(3)
8087 LET A$(8)=A$(2)
8088 LET A$(9)=A$(1)
8089 LET A$(10)="[redacted]
[redacted]"
8090 LET A$(11)="[redacted]
[redacted]"
8091 LET A$(12)=A$(11)
8092 LET A$(13)=A$(11)
8093 LET A$(14)=A$(11)
8094 LET A$(15)=A$(11)
8100 RETURN
9000 SAVE "FORMULA [redacted]"
9100 RUN
9800 PRINT "              FORMULA I"
9801 PRINT "              [redacted]"
9802 PRINT
9805 PRINT "USTED TIENE QUE TRAT
AR DE LOGRAR",,
9810 PRINT "EL MAYOR PUNTAJE POS
IBLE"
9811 PRINT
9820 PRINT "PARA ESO USTED TIENE
 QUE SALTEAR"
9821 PRINT
9825 PRINT "LOS OBSTACULOS ""[redacted]""
 ADEMAS DE ",,
9830 PRINT ,"NO SALIRSE DE PISTA
",,,
9840 PRINT "USTED MANEJARA EL AU
TO """"" CON",,
9850 PRINT ,"LOS SIGUIENTES CONT
ROLES:"
9860 PRINT
9870 PRINT "5-MUEVE AL AUTO HACI
A LA IZQ."
9880 PRINT
9890 PRINT "8-MUEVE EL AUTO HACI
A LA DERECHA"
9895 PRINT AT 21,0;"APRIETE ALGU
NA TECLA"
9900 IF INKEY$="" THEN GOTO 9900
9910 CLS
9920 PRINT "PUNTAJE:"
9925 PRINT
9930 PRINT "USTED SUMA PUNTOS A
MEDIDA QUE "
9935 PRINT
9940 PRINT "DA MAS VUELTAS AL CI
RCUITO  "
9945 PRINT
9950 PRINT "EN TOTAL SON CINCO V
UELTAS QUE"
9955 PRINT
9960 PRINT "SI LAS TERMINA OBTIE
NE 75 PUNTOS"
9965 PRINT
9970 PRINT "EXTRAS"
9975 PRINT
9980 PRINT "USTED AUMENTA SU PUN
TAJE SI COME"
9985 PRINT
9986 PRINT "LOS ""$"" QUE HAY EN
 LA PISTA"
9987 PRINT
9988 PRINT "USTED TIENE SOLO TRE
S AUTOS "
9989 PRINT
9990 PRINT "NO LOS PIERDA"
9991 PRINT "                     BU
ENA SUERTE"
9992 PRINT AT 21,0;"APRIETE UNA
TECLA PARA EMPEZAR"
9994 IF INKEY$="" THEN GOTO 9994
9996 CLS
9997 RAND
9998 RETURN
```

Los usuarios de las Sinclair, CZ o TK pueden plantear sus dudas y sugerencias en esta sección. Deben escribirnos a Cerrito 1320, piso 1°, Buenos Aires (1010). Anticipándonos a las preguntas, contestamos algunos interrogantes más comunes.

¡Cuidado!

Tengo un Spectrum de 48 K y una impresora térmica; el problema surge cuando, al querer conectar la impresora para realizar alguna operación con ella, el ordenador queda bloqueado, la impresora se pone en funcionamiento sin parar y como broche de oro pierdo el programa que quería imprimir. ¿Es normal que esto ocurra?

Es perfectamente lógico que te ocurra eso, lo raro quizá sería que no te ocurriera algo más (como por ejemplo que el problema no pase al desconectar y volver a conectar el ordenador). Hay que tener mucho cuidado con las conexiones que se efectúan en el port de expansión. En el caso de querer imprimir algo y no tener conectada la impresora, se deberá salvar en cinta el programa, con el ordenador sin alimentación conectar la impresora, luego conectar la alimentación y por último cargar el programa.

Break

¿Cómo me puedo introducir en un programa que lleva un POKE 23659,0 cuando se está ejecutando?, ¿cómo lo puedo copiar?

Una vez que está ejecutado no es cosa fácil romper un programa que lleva este tipo de protección, pero si al cargarlo hacés MERGE "" en lugar de LOAD "", el programa no se autoejecutará, por lo que se podrá buscar la línea donde se efectúa el POKE y, tras suprimirla, tendrás una versión sin proteger que podrás salvar en cinta de la forma habitual.



Amplificación

Desearía saber si al Spectrum es posible conectarle un amplificador normal u corriente, esto es: mono, de 5 watts de salida y una sensibilidad de entrada de 100 mv sobre 250 kohms sin perjuicio para el mismo.

No debe existir ningún problema si se utiliza la toma de EAR para ello, ni el ordenador ni el amplificador pueden resultar dañados.

Errata

Escriban un programa que contenga un bucle en el Spectrum. Por ejemplo:

10 PRINT "A": GOTO 10

Corran el programa, al primer "SCROLL?", CAPS y SYMBOL SHIFT juntos, y al aparecer en la línea 22 otra vez RUN (?), pulsen cualquier tecla... ... Y por favor, díganme qué es lo que sin querer he hecho aparecer en las pantallas siguientes.

Este efecto se debe a una errata de la ROM que hace que si pulsamos los SHIFT cuando se nos pregunta SCROLL? aparezca en la parte baja de la pantalla lo que haya en ese momento en la zona de edición (normalmente, la última orden introducida). A partir de ahí el sistema operativo puede perder el control e imprimir caracteres correspondientes a cualquier zona de la memoria, con el resultado ya conocido. El mismo efecto ocurre en la TS 2068.

Free

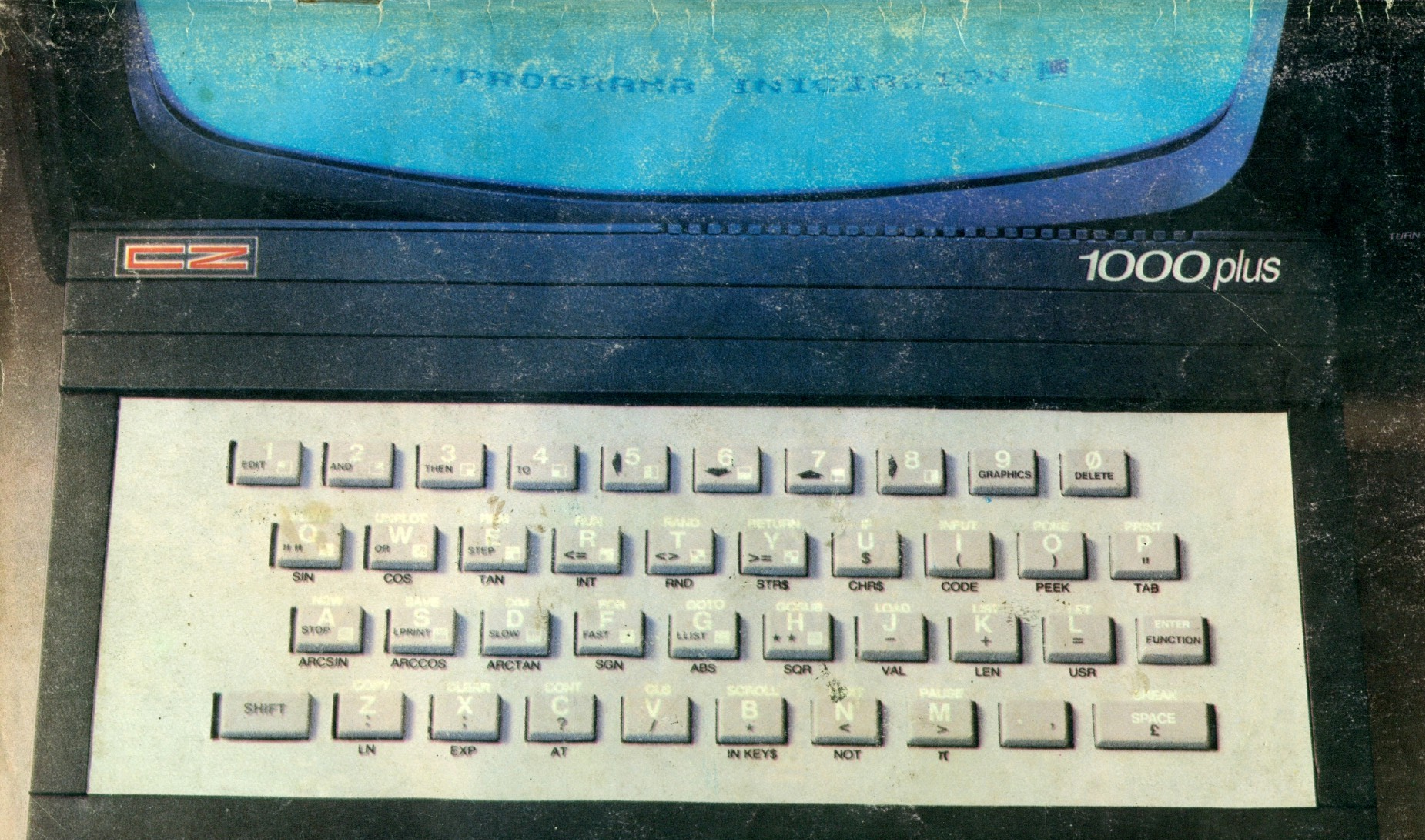
La rutina ROM 65536 - USR 7962 da la memoria que queda disponible en el Spectrum. Sin embargo, tengo algunos programas que dan el mensaje 4 (Out of memory) con resultados de esta rutina que indican que queda memoria disponible. ¿Hay alguna rutina más fiable en este sentido?

Es muy probable que, aunque en esos casos dispusieras de algo de memoria libre, la sentencia de la línea del programa que da el mensaje de error gastara, al ejecutarse, una cantidad mayor de la que había (por ejemplo una sentencia DIM, etc.).
Por otra parte se debe tener en cuenta que para ejecutar un programa en ordenador se necesita disponer de algo de espacio para los cálculos internos, almacenamiento de datos, definir variables, etc. De ahí que, aunque se pueda introducir sin problemas un determinado programa e incluso sobre algo de memoria, éste no pueda ejecutarse.

Slow-Fast

Si quiero suprimir las instrucciones SLOW y FAST de un programa para 1500 con el fin de adaptarlo a mi 2068, ¿tengo que modificar o insertar algo?

El TS 2068 trabaja siempre en modo rápido, por lo que bastará con suprimir estas sentencias del listado.



Para ingresar en el mundo de la informática